

Filière Systèmes industriels

Orientation Power and Control

Diplôme 2008

Grégory Hatt

*Interface Homme-Machine
intégrant la reconnaissance
vocale et l'analyse d'image*

Professeur

Jean-Daniel Marcuard

Expert

André Rotzetta

SI	TV	EE	IG	EST
X	X			

Confidentiel / Vertraulich

☐ oui / ja ☒ non / nein

<input checked="" type="checkbox"/> FSI <input type="checkbox"/> FTV	Année académique / Studienjahr 2007/2008	No PS / Nr. PS pc/2008/33
Mandant / Auftraggeber <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Ecole hôte	Etudiant / Student Grégory Hatt	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Ecole hôte
Professeur / Dozent Jean-Daniel Marcuard	Expert / Experte (données complètes) André Rotzetta - HES-SO Fribourg	

Titre / Titel

**Interface Homme-Machine intégrant la reconnaissance vocale
et l'analyse d'image**

Description et Objectifs / Beschreibung und Ziele

Les interfaces homme-machines conventionnelles n'utilisent qu'une petite partie des moyens de communication dont dispose l'être humain : l'opérateur donne des ordres en appuyant sur des boutons ou sur un clavier et la commande informe l'utilisateur avec des voyants, des affichages et des alarmes sonores. Techniquement, il est aujourd'hui possible d'utiliser des canaux de communication plus naturels pour l'être humain : commande vocale, détection de gestes significatifs par analyse d'image, synthèse vocale pour informer l'utilisateur, etc.

Le projet consiste à développer une interface homme machine mettant en œuvre ces canaux de communication. Elle sera implémentée sur PC et basée autant que possible sur des logiciels standards (par ex : bibliothèques intégrées à Windows XP ou Vista). Au moins deux applications de démonstration seront développées :

- La première permettra de gérer une maquette de machine-outil commandée par un automate programmable industriel.
- La seconde permettra de commander un robot de façon conviviale.

L'étudiant recherchera les applications susceptibles de profiter pleinement de ces technologies. Il étudiera notamment la faisabilité d'une interface permettant de communiquer avec des personnes malentendantes en utilisant le langage des signes.

Enfin, il évaluera les risques liés à l'utilisation de ces techniques (aspect sécurité) et les moyens de les limiter.

Signature ou visa / Unterschrift oder Visum

Resp. de la filière

Leiter des Studieng.:

Etudiant/Student:

Délais / Termine

Attribution du thème / Ausgabe des Auftrags:
01.09.2008Remise du rapport / Abgabe des Schlussberichts:
21.11.2008, 12:00Exposition publique / Ausstellung Diplomarbeiten:
28.11.2008Défense orale / Mündliche Verfechtung:
semaine/Woche 49

Interface Homme-Machine intégrant la reconnaissance vocale et l'analyse d'image

Schnittstelle Mensch-Maschine mit integrierter Spracherkennung und Bildanalyse

Objectifs

Ce projet a pour objectif principal de développer une interface Homme-Machine utilisant la reconnaissance vocale et l'analyse d'image. Les techniques de reconnaissance doivent être basées sur des logiciels standards et devront être implémentées sur PC.

Deux applications de démonstrations doivent être réalisées en utilisant ces techniques.

Un objectif secondaire consiste à faire une étude sur les applications qui pourraient profiter pleinement de ces technologies et d'en évaluer les risques d'utilisation.

Résultats

L'interface a été réalisée de manière complète et conviviale. Une certaine souplesse est donnée à l'utilisateur dans l'emploi des outils développés. Ces outils sont fonctionnels.

La première application a été réalisée durant le travail de semestre.

La seconde permet de piloter un robot Delta. L'application fonctionne bien.

Ces technologies ont de l'avenir dans nos relations avec les machines et plusieurs exemples de domaines d'applications ont été trouvés.

Mots-clés

Interface Homme-Machine, analyse d'image, webcam, reconnaissance vocale, micro, synthèse vocale, robot Delta, pointeur laser, chocolat.

Zielsetzungen

Ziel dieses Projektes ist es, eine Schnittstelle Mensch-Maschine zu entwickeln mit Spracherkennung und Bildanalyse. Die Erkennungs-Techniken für die Realisierung müssen auf Standardsoftware basieren und auf dem PC implementiert werden.

Zwei Demo-Anwendungen müssen mit dieser Technik realisiert werden.

Ein zweites Ziel besteht darin, eine Studie zu machen über mögliche Anwendungen, die von diesen Technologien profitieren könnten und die Risiken deren Nutzung abschätzen.

Resultate

Die Benutzeroberfläche wurde vollständig und benutzerfreundlich erstellt. Eine gewisse Flexibilität wird dem Benutzer überlassen bei der Verwendung der entwickelten Instrumente. Diese Instrumente sind funktionsfähig.

Die erste Anwendung wurde während der Semester-Arbeit realisiert.

Die zweite erlaubt die Steuerung eines Delta-Roboters. Die Anwendungen funktionieren einwandfrei.

Diese Technologien haben gute Zukunftsaussichten für unsere Beziehung zur Maschine und mehrere Beispiele für derartige Anwendungsbereiche sind gefunden worden.

Schlüsselwörter

Schnittstelle Mensch-Maschine, Bildanalyse, Webcam, Spracherkennung, Mikro, Sprachsynthese, Delta-Roboter, Laser Pointer, Schokolade

Table des matières

1	Introduction	8
1.1	Présentation générale	8
1.2	Organisation du projet	8
1.3	Cahier des charges	9
2	Rappel du projet de semestre	10
3	Recherches et analyses	11
3.1	Langage des signes	11
3.1.1	Présentation du langage	11
3.1.2	Traduction voix/langage des signes	13
3.1.3	Traduction langage des signes/voix	13
3.2	Analyse des risques	14
3.2.1	Risques de la reconnaissance vocale	14
3.2.2	Risques de l'analyse d'image	15
3.3	Recherche d'une application	15
4	Outils pour l'interface	17
4.1	Analyse d'image	17
4.1.1	Résolution d'analyse	18
4.1.2	Détection des couleurs	19
4.1.3	Gestion de la lumière	22
4.1.4	Reconnaissance du pointeur	24
4.2	Reconnaissance vocale	25
4.2.1	Ajout du composant	25
4.2.2	Déclaration du dictionnaire	26
4.2.3	Reconnaissance des mots	27
4.3	Synthèse vocale	27
4.3.1	Ajout du composant	28

4.3.2	Utilisation de l'objet	28
4.4	Base de données	28
4.4.1	Déclaration des objets	29
4.4.2	Parcours d'une base de données	29
4.4.3	Ajout d'une nouvelle entrée	30
4.4.4	Suppression d'une entrée	30
4.5	Communication	31
4.5.1	Ajout du composant	31
4.5.2	Gestion de la communication	32
4.5.3	Réception de données	32
4.5.4	Envoi de données	32
4.5.5	Protocole personnel	32
5	Présentation de l'interface	34
5.1	Introduction	34
5.2	Présentation de l'aspect visuel	34
5.2.1	Aspect général	35
5.2.2	Menu principal	37
5.2.3	Fenêtre du mode automatique	38
5.2.4	Fenêtre du mode pas-à-pas	39
5.2.5	Fenêtre du mode d'enregistrement de produit	40
5.3	Paramètres de configuration	41
5.3.1	Aspect du menu	41
5.3.2	Gestion Vidéo	42
5.3.3	Gestion des couleurs	45
5.3.4	Reconnaissance vocale	47
5.3.5	Réseau	48
6	Présentation de la réalisation pratique	49
6.1	Remplissage automatique	50
6.1.1	Principe	50
6.1.2	Structure de la programmation	51
6.2	Remplissage pas-à-pas	52
6.2.1	Principe	52
6.2.2	Schéma de programmation	53
7	Conclusions	54
8	Perspectives	55
8.1	Améliorations	55
8.2	Utilisations réelles	55
8.2.1	Gestion de stock	55
8.2.2	Pilotage de machine de chantier	56
9	Références	57

10 Remerciements	58
11 Annexes	59
11.1 Plans du réservoir de chocolats	59
11.2 Contenu du CD	65
11.2.1 Rapport	65
11.2.2 Programme de l'interface - Visual Basic	65
11.2.3 Programme du robot - TurboPascal	65
11.2.4 Outils Microsoft	65
11.2.5 Autres	65

Table des figures

3.1	Alphabet dactylogique français	13
4.1	Représentation d’une boucle FOR pour parcourir un tableau	18
4.2	Représentation d’une boucle FOR avec fonction STEP pour parcourir un tableau plus rapidement. Seuls les points foncés sont analysés	19
4.3	Représentation des couleurs, synthèse soustractive à gauche et additive à droite	19
4.4	Exemple de tableau de pixel référence	20
4.5	Principe de détection par différence des paramètres RVB	21
4.6	Comparaison des corrections de luminosité	23
4.7	Détection du pointeur laser sur la zone de commande	24
4.8	Composant de reconnaissance vocale	25
4.9	Code permettant de déclarer le dictionnaire et de l’enregistrer . . .	26
4.10	Déclaration de la fonction de reconnaissance vocale et début de fonction <i>SELECT CASE</i>	27
4.11	Composant de synthèse vocale	28
4.12	Boucle de parcours de base de donnée	30
4.13	Composant de communication TCP/IP	31
5.1	Présentation générale de l’interface	35
5.2	Partie supérieure de la zone de visualisation	36
5.3	Barre d’informations de l’interface	37
5.4	Boutons de la page d’accueil et leur description	37
5.5	Eléments de la page de composition automatique	38
5.6	Représentation simplifiée de la page du mode pas-à-pas	39
5.7	Représentation simplifiée de la page du mode de création d’un nouveau produit	40
5.8	Menu configuration : Général	41
5.9	Menu configuration : Gestion de la vidéo	42
5.10	Paramètres de l’analyse de la vidéo	43

5.11	Menu configuration : Gestion des couleurs	45
5.12	Menu configuration : Reconnaissance vocale	47
5.13	Menu configuration : Réseau	48
6.1	Structure du programme du mode automatique	51
6.2	Structure du programme du mode pas-à-pas	53

CHAPITRE 1

Introduction

1.1 Présentation générale

Mon travail de diplôme consiste à développer une interface homme-machine se basant sur les outils de reconnaissance vocale de Windows XP, ainsi que d'une reconnaissance d'image avec une webcam USB. Cela permettra de communiquer entre un automate quelconque et un ordinateur standard équipé d'un micro et d'une webcam.

De plus, une réalisation concrète correspondant à une utilisation industrielle devra être développée.

1.2 Organisation du projet

J'ai partagé mon travail de diplôme en quatre étapes principales :

Analyses des possibilités d'applications. L'analyse d'image et la reconnaissance vocale offrent de multiples possibilités.

Développement des outils. Recherches et développement des possibilités d'utilisation de la webcam et du micro.

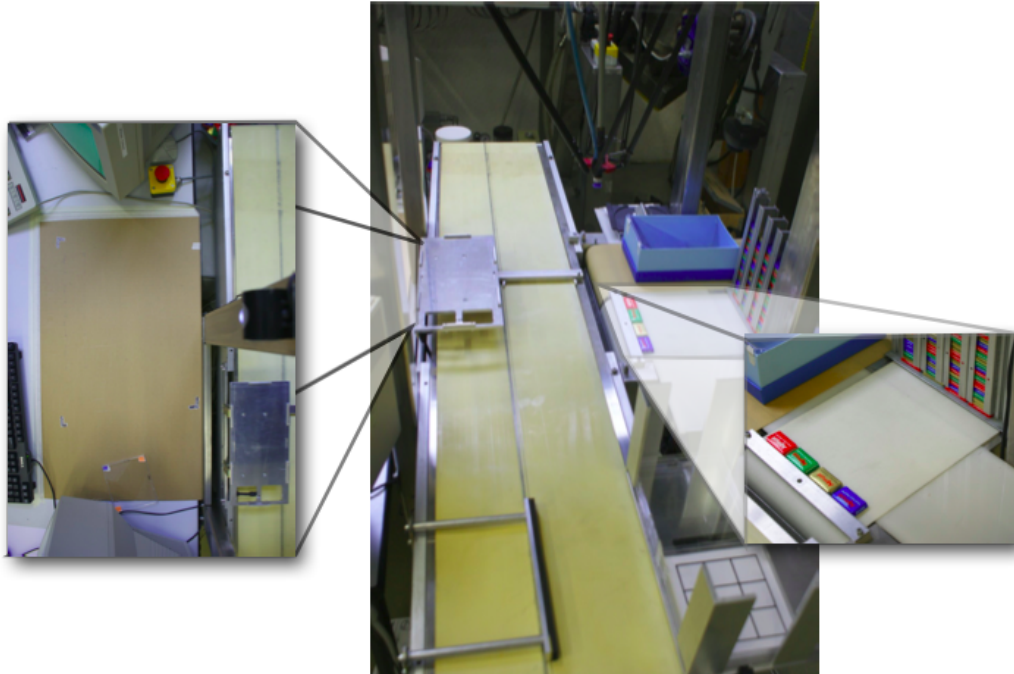
Développement de l'interface. Réalisation d'un programme offrant une interface intuitive.

Réalisation d'une application de démonstration. Gestion d'un robot à l'aide de l'interface et des outils développés

1.3 Cahier des charges

Le travail de semestre m'a permis d'identifier les possibilités de réalisations qui pourraient correspondre à une utilisation industrielle. Pour ce travail de diplôme, un cahier des charges a été établi afin de réaliser une application autour du robot Delta :

- Chercher des possibilités de débouchés et d'utilisations industrielles ou sociales.
- Développer les outils de détection de couleurs.
- Développer une méthode de reconnaissance vocale efficace.
- Réaliser une application permettant de piloter le robot Delta.



CHAPITRE 2

Rappel du projet de semestre

Durant mon travail de semestre, j'ai analysé les différentes possibilités d'utilisation d'une webcam et d'un micro comme interface afin de piloter un automate. J'avais également réalisé des tests sur les différents outils mis à disposition par Microsoft.

Le travail avec la reconnaissance vocale s'était bien déroulé. Malheureusement, Microsoft ne met à disposition sur XP que la reconnaissance de l'anglais, du chinois simplifié et du japonais. Les mots devaient être écrits dans un fichier pour être reconnus. J'avais tenté d'enregistrer les mots en phonétique, de sorte qu'en les prononçant en anglais, le son corresponde au mot en français. Le principe était trop limité.

Je n'avais pas beaucoup poussé l'analyse d'image, car j'en avais déjà fait au cours d'un travail précédent et je savais que c'était réalisable et fonctionnel. J'avais donc simplement réalisé une détection de présence.

Un autre problème consistait à communiquer entre l'ordinateur et l'automate programmable Modicon. Pour cela, l'automate est configuré pour comprendre le protocole Modbus. Ce protocole est un standard dans la communication avec les automates. Il a fallu que le programme Visual Basic soit configuré pour émettre des signaux par le port série.

Au final, j'avais réalisé une petite application avec des vérins pneumatiques que je contrôlais à la voix, seulement si je me trouvais devant la webcam. Cela dépassait le cadre de mon travail de semestre, mais cela m'a permis de prendre de l'avance sur mon travail de diplôme.

En conclusion, je terminais en disant que les interfaces homme-machine étaient un sujet très intéressant et que je me réjouissais de développer une telle interface.

CHAPITRE 3

Recherches et analyses

Les différents systèmes existants d'interface entre l'homme et la machine ont été présentés dans mon travail de semestre. Comme précisé dans le rapport de cet ancien travail, les domaines touchés par des interfaces innovantes sont principalement la recherche et le monde du jeu vidéo.

Seulement, ce ne sont pas les seuls domaines qui pourraient bénéficier de ces technologies. Il faudrait voir ce qui est possible de réaliser afin d'aider les sourds et malentendants par exemple.

La sécurité est aussi un domaine qui pourrait bénéficier de ces techniques. Seulement, dans ce cas, il faut s'assurer que le système soit suffisamment sûr pour se permettre de lui attribuer des tâches sensibles.

3.1 Langage des signes

Un des points du travail de diplôme est d'analyser les possibilités d'une interface Homme-Machine pour assister les personnes malentendantes et utilisant le langage des signes.

Par exemple, à l'aide d'un traducteur langage des signes / voix, une personne malentendante pourrait communiquer avec une personne ne connaissant pas le langage des signes.

3.1.1 Présentation du langage

Le langage des signes est une langue visuelle et tri-dimensionnelle. Elle est utilisée par les sourds et les mal-entendants depuis 1760. C'est à cette période que le système fut imaginé.

Selon le site *langage-des-signes.com*¹, la formation des signes utilise 5 éléments qui se combinent :

La configuration : Cela concerne la forme de la main. La ligue française des signes en a retenu 60.

L'orientation de la main : les mains peuvent être orientées paumes vers le bas, le haut, l'une vers l'autre, etc. Les bras peuvent être horizontaux, verticaux, obliques, etc.

L'emplacement : l'endroit où les signes se font soit une quinzaine d'endroit sur le corps et trois principaux dans l'espace. Par exemple, le signe peut se placer sur la bouche, les yeux, le bras, l'estomac, la paume, etc.

Le mouvement : un ou plusieurs mouvements peuvent être utilisés. Ils peuvent être simples (une main qui descend) ou complexes (deux mains qui se rapprochent avec les doigts qui vibrent). Les mouvements de tête peuvent aussi entrer en jeu. Différents éléments peuvent intervenir pendant un mouvement dont le trajet, la direction et la vitesse.

L'expression du visage : donne un sens à un signe isolé et est fondamentale dans la construction d'une phrase.

D'après ces informations, j'ai constaté que le langage des signes était plus complexe que ce que j'imaginai. En effet, le nombre de paramètres à prendre en compte est très grand. La traduction du langage des signes se fait selon le contexte de la phrase et les expressions du visage. Ces deux notions sont particulièrement difficiles à analyser pour une machine.

Dans les langues parlées, il existe des dialectes différents qui se ressemblent peu et il en est de même pour le langage des signes. Les langues des signes se basent toutes sur les langues orales des pays où elles sont pratiquées. Il existe donc une grande quantité de signes et il est quasiment impossible de les connaître tous.

J'ai donc essayé de revoir mes objectifs en analysant uniquement l'alphabet dactylogique. Il est réalisé avec une main dans la plupart des pays d'Europe et avec deux mains en Angleterre, en Irlande et en Australie².

¹<http://www.langage-des-signes.com>

²Les alphabets dactylogique de différents pays sont visibles sur internet à l'adresse suivante : <http://picasaweb.google.com/WorldSignLanguage/DactylogieHandalphabet>



FIG. 3.1: Alphabet dactylologique français

La détection des signes de l'alphabet reste tout de même un problème, car certains signes sont effectués avec des mouvements complexes. Par exemple, le Z en français est fait en écrivant la lettre dans l'air.

3.1.2 Traduction voix/langage des signes

La traduction dans ce sens est réalisable. En effet, l'outil de reconnaissance vocale permet de comprendre l'utilisateur. Si la base de donnée est complète, il est possible de comprendre toutes les lettres. D'une fois qu'un caractère est compris, il suffit d'afficher la lettre à l'écran.

Avec une base de donnée plus poussée, il est possible d'envisager de reconnaître des mots. Si un mot est compris, une vidéo est affichée montrant le signe correspondant.

3.1.3 Traduction langage des signes/voix

La synthèse vocale fonctionnant correctement, prononcer un mot ou une lettre par l'ordinateur ne représente pas un problème. Par contre, l'analyse d'un signe est particulièrement complexe.

Il est difficile d'analyser les doigts séparément. Il faudrait faire des comparaisons avec des formes enregistrées et bien détecter les formes. Une solution serait de donner à l'utilisateur des gants dont chaque doigt est d'une couleur différente.

3.2 Analyse des risques

Le fait d'avoir 5 sens qui fonctionnent prive parfois l'être humain d'un certain recul sur l'immense capacité de son cerveau. Par exemple la vue et l'ouïe sont des outils très performants et, à force de les utiliser tous les jours, nous avons l'impression qu'ils ne nous demandent aucun effort. Par contre, lorsqu'il faut apprendre à une machine à analyser de la voix ou une image, le nombre de paramètres à prendre en compte et les démarches sont impressionnants.

La machine tente de copier l'homme, c'est-à-dire que le meilleur moyen de reconnaître quelque chose est de le comparer à quelque chose d'équivalent et que nous connaissons bien. Par exemple, pour reconnaître un mot, elle va le comparer à une banque de sons enregistrés et dont elle connaît la signification.

Dans le domaine de la sécurité dans un milieu industriel, il arrive souvent que les erreurs soient humaines. Alors comment faire pour améliorer une machine supervisant cette sécurité si l'homme lui-même n'y arrive pas toujours ?

3.2.1 Risques de la reconnaissance vocale

Dans le cadre de mon travail de diplôme, j'ai été confronté au cas où j'ai dû protéger mon programme afin qu'il n'effectue pas des actions non désirées après avoir compris un mot. En effet, la reconnaissance vocale avait tendance à reconnaître à tout moment et d'agir en conséquence. Pour diminuer ce risque de mauvaise compréhension du programme, j'ai opté pour plusieurs solutions selon l'ordre donné et l'action qui en découlait :

Mots d'ordre - Cette technique concerne les mots qui peuvent être donnés à tout moment, mais dont la fonction pourrait être gênante si elle était mal interprétée. Elle consiste à placer deux autres mots de commande avant celui désiré. Dans mon cas, j'ai décidé que les mots *Command* et *Start* étaient réservés à cet effet. Donc avant qu'un ordre soit prononcé, le système aura du reconnaître les mots de commande, sinon l'ordre ne sera pas exécuté.

Situation - C'est la deuxième technique, elle concerne les mots qui ne sont prononcés que dans des situations particulières. L'accès à la fonction qui est rattachée à un mot est donc protégé par un test logique qui détermine si la situation est adéquate. Par exemple, une commande qui ferait arrêter un robot n'a pas besoin d'être reconnue si le robot est arrêté. Ou encore, si l'interface est dans le menu principal et que l'ordre compris concerne un autre menu, il n'a pas de raison d'être écouté.

Autorisation - La dernière technique consiste à laisser comprendre des mots d'ordre uniquement à des moments choisis par le développeur. L'écoute serait ainsi coupée et redémarrée en fonction de l'action en cours.

Ces trois méthodes se sont révélées efficaces, les bruits ou discussions ambiantes n'ont plus eu d'effet sur le programme. Les risques d'une mauvaise compréhension ont ainsi été diminués.

Le risque zéro n'existe malheureusement pas et malgré ces méthodes, il arrive que l'ordre donné ait un son tellement proche d'un autre connu que la différence n'est pas toujours faite. Afin d'éviter de mauvaises surprises de ce genre, c'est au développeur de faire attention à ce que deux ordres pouvant être donnés dans un même contexte ne se ressemblent pas.

3.2.2 Risques de l'analyse d'image

La reconnaissance de formes ou de couleurs est dépendante d'un apprentissage. Si cette phase n'a pas été effectuée correctement ou que certains paramètres de l'environnement ont changé, l'analyse n'est plus fiable. Par exemple, si la webcam est déplacée, alors les références pré-établies n'auront plus aucun sens.

Il faut donc prévoir dans le programme une phase de calibrage, afin de s'assurer du bon fonctionnement de l'analyse d'image. Cette phase doit s'effectuer une fois au début et être accessible dans un menu adéquat. Si certains paramètres ne demandent pas de calibrage mais un enregistrement, ils doivent être sauvegardés dans un fichier externe au programme. Ainsi en cas de problème du programme, ces valeurs ne seront pas affectées.

Pour mon travail de diplôme, la phase de calibrage a été imaginée, mais, par manque de temps, elle n'a pas pu être testée. Cette fonctionnalité est décrite dans le chapitre *Perspectives*.

3.3 Recherche d'une application

Afin de tester ces différents outils, leurs risques et leurs limites, une application devait être développée. L'interface devait donner à l'utilisateur la possibilité d'interagir avec la machine par la voix et l'image.

Concernant la voix, les fonctions ne demandaient pas une grande réflexion. Les ordres vocaux devaient simplement remplacer les boutons. La synthèse vocale remplaçait les voyants ou autres messages visuels.

Par contre, il a fallu trouver une application où l'analyse d'image avait une fonction plus évoluée qu'une simple détection de présence.

J'ai appris au début de mon travail de diplôme qu'un robot Delta était disponible. Ce type de robot est conçu pour faire des actions simples appelées *Pick*

and Place. Ce nom anglais définit bien la fonction du robot qui est de prendre et de placer. Généralement utilisé dans des fabriques d'emballage, ces robots sortent tout droit du monde de l'industrie.

Il a fallu trouver une utilisation pratique à l'analyse d'image. L'idée qui en est ressortie est de détecter la couleur de l'objet qui va être mis dans la boîte ainsi que de donner l'ordre visuellement de prendre et de poser cet objet. C'est donc ces objectifs que je me suis fixés pour mon interface :

- Donner des ordres vocaux au robot Delta.
- Recevoir des indications sonores sur l'évolution du programme.
- Détecter des couleurs et en apprendre de nouvelles.
- Détecter des positions selon des gestes ou des signaux.

CHAPITRE 4

Outils pour l'interface

Avant de créer une interface homme-machine efficace, il faut développer les différents outils qui seront utilisés. Ces outils seront la base de cette interface. Ils ont été développés pour correspondre aux objectifs que je m'étais fixés au chapitre précédent.

4.1 Analyse d'image

La reconnaissance d'image est basée sur une capture d'image et l'analyse de celle-ci pixel par pixel. En effet, nous pouvons comparer une image numérique à un tableau en deux dimensions de couleurs. Dans le logiciel VisualBasic, deux DLL sont utilisées pour capturer le signal vidéo depuis une webcam USB :

avicap32.dll : Elle permet de faire une capture vidéo ; normalement elle est installée d'origine avec Microsoft XP. Elle se trouve dans le dossier :
"C :\Windows\System32"

La fonction utilisée est : `capCreateCaptureWindow`, elle permet de créer un objet qui permettra d'accéder à la capture de la vidéo en envoyant des messages avec la DLL suivante.

user32.dll : Cette librairie fait également partie des bases de Windows, elle se trouve dans le même dossier que la `avicap32.dll`. Elle a de multiples fonctions, mais dans notre cas, elle permet de transmettre des messages à l'autre DLL afin de démarrer ou arrêter la capture, de déterminer quel driver est nécessaire, etc.

C'est par cette DLL que l'on va transmettre les informations pour la capture. La fonction est : `SendMessage`.

Les paramètres de ces fonctions se trouvent dans le code Visual Basic dans le module type.bas. Par contre, il y a très peu d'explications fournies pour ces DLL.

4.1.1 Résolution d'analyse

Parcourir un tableau de pixels peut prendre du temps et dans un système industriel, le temps c'est de l'argent. Afin d'obtenir une information d'une image, il n'est pas toujours nécessaire d'observer tous les points en détail. Il est possible d'accélérer le parcours d'une image en diminuant la résolution d'analyse.

Par exemple, avec une résolution de 320 par 240 pixels, l'image devrait être testée 76'800 fois. En ne testant qu'une colonne sur deux et qu'une ligne sur deux, le nombre de test passe à 19'200, soit quatre fois plus rapide. Selon l'importance de l'objet à analyser dans l'image, il est possible de ne tester qu'une colonne et qu'une ligne sur 5 voir 10, ce qui augmente la vitesse respectivement de 25 à 100 fois.

En Visual Basic, le code pour parcourir un tableau est composé de deux boucles "FOR" représentant les colonnes et les lignes. Le code est le suivant :

```
For X = 1 To 320
  For Y = 1 To 240
    tableau(X, Y) = Picture(X, Y)
  Next Y
Next X
```

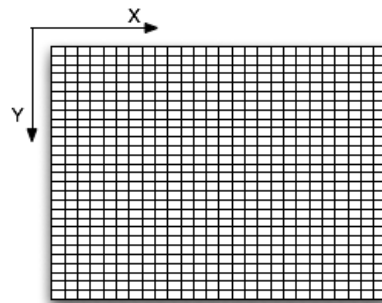


FIG. 4.1: Représentation d'une boucle FOR pour parcourir un tableau

Afin de régler la résolution d'analyse, il est possible de rajouter à la déclaration de la boucle la fonction "STEP". Cette dernière définit l'incrément à ajouter à chaque exécution de la boucle.

```
For X = 1 To 320 Step 5
  For Y = 1 To 240 Step 5
    tableau(X, Y) = Picture(X, Y)
  Next Y
Next X
```

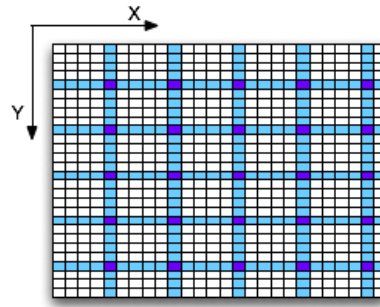


FIG. 4.2: Représentation d'une boucle FOR avec fonction STEP pour parcourir un tableau plus rapidement. Seuls les points fongés sont analysés

4.1.2 Détection des couleurs

Dans le monde de l'informatique, le code le plus répandu est le code dit RVB, pour Rouge - Vert - Bleu. Chacune des couleurs est codées sur 8 bits, c'est-à-dire de 0 à 255, généralement noté en hexadécimal. Dans le système de couleurs additives, elles sont les trois couleurs primaires. En couplant ces couleurs par deux, nous obtenons du jaune, du cyan et du magenta, les trois couleurs secondaires.

Afin de composer les autres couleurs du spectre de lumière, il faut mélanger ces trois primaires en plus ou moins grande quantité. C'est cette quantité que définit le code RVB. Par exemple, pour avoir du rouge pur, il faut 255 de rouge, 0 de vert et 0 de bleu, soit le code #FF0000.

En couplant ces couleurs par deux, nous obtenons du jaune, du cyan et du magenta, les trois couleurs secondaires.

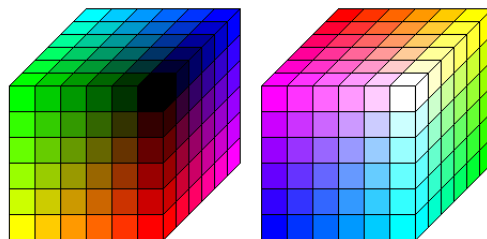


FIG. 4.3: Représentation des couleurs, synthèse soustractive à gauche et additive à droite

Afin de détecter au mieux les couleurs, plusieurs méthodes ont été étudiées et sont décrites dans les sous-chapitres suivants.

Méthode comparative

Afin de connaître la couleur du pixel observé, on compare son code couleur à une autre image modèle contenant toutes les couleurs. Cette méthode exige que l'on teste la couleur de chaque pixel. Comme expliqué précédemment, une image est un grand tableau de couleur. Donc en parcourant toutes les cases et en testant leur couleur, il est possible de les comparer.

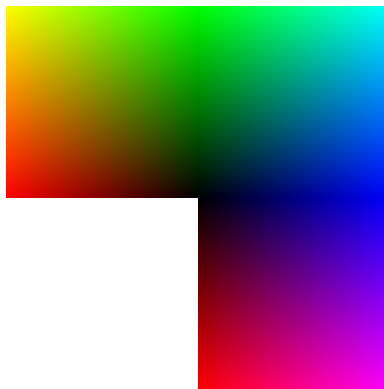


FIG. 4.4: Exemple de tableau de pixel référence

Le pixel de l'image modèle qui correspond à notre couleur a une coordonnée précise dans l'image, une colonne et une ligne. Selon ces coordonnées, il sera possible, d'après des zones pré-établies, de définir de quelle couleur il s'agit.

Résultats : Cette méthode n'est pas efficace. Dans un premier temps, elle exige de parcourir l'ensemble de l'image modèle, ce qui prend du temps.

Ensuite, dans le cas où la luminosité change, certaines déclinaisons de couleurs ne seraient pas couvertes sans une image modèle immense, ce qui rallongerait encore le temps de recherche.

Détection par proximité des paramètres RVB

Ensuite, dans la seconde méthode, lorsqu'une couleur est recherchée, chaque paramètre de chaque pixel d'une image est décortiqué. Si les valeurs des paramètres R, V et B de la couleur cherchée se rapprochent des paramètres R, V et B du pixel analysé avec une marge plus ou moins grande, alors les couleurs sont considérées comme égales.

Résultats : Cette méthode permet d'enregistrer facilement une couleur par un utilisateur. Un seul clic sur un pixel d'une image permet d'en connaître sa composition.

En variant plus ou moins la taille de la marge de détection, cette dernière sera plus correcte en cas de changement de luminosité. Malheureusement, cette méthode a ses limites. De gros changements de luminosité empêchent toute détection valable.

Détection par différences des paramètres RVB

La dernière méthode est un peu plus complexe. Il faut déterminer les écarts existants entre les différents paramètres RVB d'un pixel. Ainsi, selon la couleur dominante et l'importance de celle-ci, il est possible de détecter les 3 couleurs primaires.

Dans un second temps, les couleurs secondaires peuvent également être détectées. Si deux paramètres ont des valeurs proches, mais éloignées du troisième, cela concerne une des couleurs secondaires.

Ces notions de "proches" et "d'éloignés" se rapportent à la différence en quantité de couleur de chaque paramètre. Elles peuvent être représentées en pour-cent ou en nombre de points de 0 à 255.

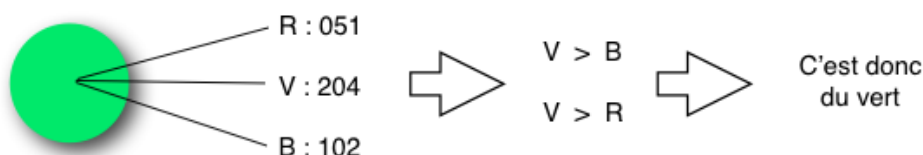


FIG. 4.5: Principe de détection par différence des paramètres RVB

Résultats : Cette méthode permet surtout de détecter les couleurs primaires et dans une certaine mesure les couleurs secondaires. Malgré cette limite, c'est la méthode que j'ai retenue, car elle permet une plus grande souplesse dans la luminosité sans toutefois être parfaite.

Perspectives : Il serait peut-être possible d'approfondir cette méthode pour détecter d'autres couleurs que les six principales. Par exemple, en enregistrant les écarts entre chaque paramètre, en permettant une légère marge de précision de

plus ou moins quelques points. Je n'ai pas eu le loisir de tester cette technique d'apprentissage d'une couleur car elle présente une plus grande complexité pour l'utilisateur et cela ne convient pas à l'interface développée.

4.1.3 Gestion de la lumière

Comme expliqué précédemment dans ce rapport, la lumière ambiante est un mauvais allié dans la détection de couleur et donc dans l'analyse d'image en général. Certains codages de couleurs contiennent un paramètre dédié à la luminosité, ce qui n'est pas le cas du système RVB. La luminosité s'obtient en faisant la moyenne des trois paramètres de couleurs. Un point lumineux s'approchera de 255 et inversement un point sombre tendra vers 0.

Donc pour augmenter la luminosité d'une image, il faut que chaque paramètre de chaque pixel soit multiplié par un facteur de correction. Afin d'obtenir la meilleure détection des couleurs, trois méthodes de calculs ont été testées afin d'obtenir le meilleur facteur.

Pour comparer l'efficacité des méthodes, une détection des couleurs par la différences des paramètres RVB a été faite.

Mise à niveau de chaque point

La luminosité moyenne de l'image est mesurée en prenant une série de points dans l'image avec une résolution assez grande et en faisant la moyenne générale.

Ensuite, l'image est parcourue à nouveau et la luminosité de chaque pixel est analysée. Le rapport de la luminosité du point sur la luminosité de l'image donne le facteur de correction. Chaque paramètre RVB est ensuite multiplié par ce facteur afin d'élever ou d'abaisser la luminosité du pixel à la même valeur que la luminosité de l'image.

Résultats : L'analyse des couleurs est bien améliorée. Les zones plus sombres ont été révélées et rendues détectables par la méthode de différences des paramètres RVB. Si la luminosité de l'image est faible, la méthode est inutile, car la valeur de la luminosité étant également basse, cela ne nous sert pas à grand chose. Par contre, si cette méthode est utilisée sur une image qui ne contient que de petites zones sombres, alors le résultat est satisfaisant.

Amélioration : Il est possible d'améliorer cette méthode en fixant la valeur de luminosité à atteindre. Ainsi, les images sombres sont également exploitables.

Augmentation des basses lumières

Suite aux tests de la première méthode, j'ai remarqué qu'en diminuant la luminosité des pixels plus clairs, je prenais le risque de limiter la détection.

La solution est donc d'augmenter la luminosité uniquement des pixels plus foncés qu'une valeur fixée. Cette valeur est déterminée par plusieurs tests à différentes luminosités. Je l'ai donc fixée à 160 unités sur 255, soit environ 63% de luminosité.

Résultats : Dans le cas où l'image est sombre, les résultats obtenus sont les mêmes que pour le premier système avec la valeur prédéfinie. Dans ce genre de cas, il n'est donc toujours pas possible d'obtenir un résultat satisfaisant. Par contre, pour une situation normale, avec peu de zones d'ombres, cette méthode à l'avantage d'éclaircir uniquement ces zones et d'obtenir ainsi un résultat plus propre.

Augmentation à une haute valeur

Afin d'obtenir une détection dans toutes les situations, la luminosité de chaque pixel est augmentée à une valeur de 200 unités, soit à environ 80% de luminosité.

Résultats : Le résultat obtenu n'est pas satisfaisant. En effet, à faible luminosité, les écarts entre les valeurs des paramètres RVB sont tellement petits que même en les augmentant, les différences ne sont pas assez marquées pour rendre reconnaissables les couleurs.

Conclusions

Dans la figure suivante, les trois principes sont illustrés. La première image est d'origine et les suivantes sont respectivement par mise à niveau, par augmentation des basses lumières et augmentation à une haute valeur.

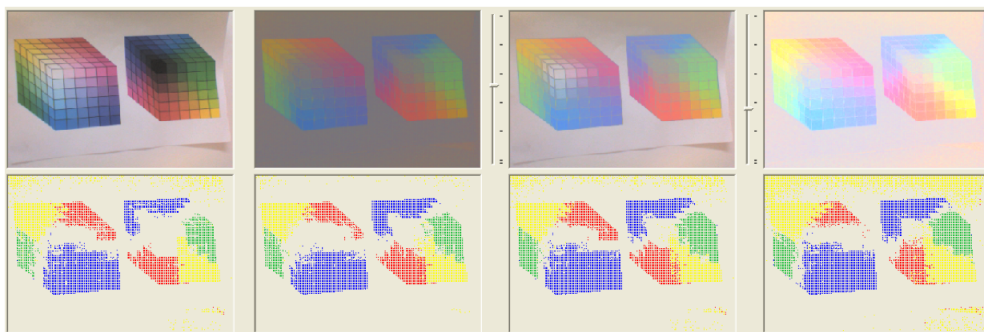


FIG. 4.6: Comparaison des corrections de luminosité

En définitive, les deux premières méthodes sont tout à fait exploitables. La deuxième offre l'avantage d'un résultat plus propre et plus rapide, étant donné que moins de pixels sont corrigés. Le troisième principe n'a pas été retenu, car il avait tendance à trop abîmer les couleurs.

C'est donc la méthode d'augmentation des basses lumières que j'ai retenue, car c'était celle dont les résultats étaient les plus convaincants. Il faut tout de même rappeler que ce système n'est pas infallible si l'image est vraiment trop sombre, soit avec une luminosité moyenne inférieure à 15%.

4.1.4 Reconnaissance du pointeur

Le dernier outil développé est en fait un dérivé de la détection de couleur. Cet outil permet de détecter le pointeur lumineux. La détection de deux types de pointeurs a été programmée.

Pointeur DEL - Afin de détecter la position d'une lampe de poche à Diode Electroluminescence (DEL), la luminosité des pixels d'une image est analysée. Si la valeur de la luminosité dépasse une très haute valeur, alors il s'agit du pointeur, comme par exemple 250, soit 98%.

Pointeur laser - Le pointeur laser envoie une lumière très rouge. Il faut donc analyser l'image et déterminer si un point est d'un rouge très pur, c'est-à-dire un paramètre R très grand et des paramètres V et B faibles.

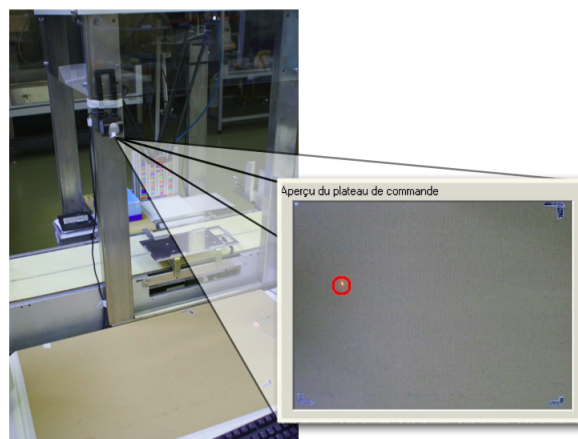


FIG. 4.7: Détection du pointeur laser sur la zone de commande

Lorsqu'un point est détecté comme pointeur, il est cerclé de rouge afin que l'utilisateur le sache.

4.2 Reconnaissance vocale

Microsoft met à disposition, sur son site internet, des kits de développement permettant de faire de la reconnaissance vocale. Il y a plusieurs exécutables à télécharger :

actcnc.exe : C'est le moteur de reconnaissance vocale. Il permet de reconnaître des mots qui doivent être enregistrés et déclarés dans ce que Microsoft appelle un dictionnaire. Le moteur ne fonctionne malheureusement qu'en anglais, en chinois simplifié et en japonais. L'anglais a bien entendu été choisi.

speechSDK51.exe : C'est le kit de développement lui-même. Il contient les routines de gestion de la parole pour Windows XP. Il permet d'avoir les différents composants sous Visual Basic permettant la reconnaissance vocale.

spchcpl.exe : Ceci est le panneau de configuration permettant d'approfondir la reconnaissance de la voix d'une personne et de son environnement.

4.2.1 Ajout du composant

Pour ajouter un objet de reconnaissance vocale dans Visual Basic, il faut faire un clic droit dans la barre d'outils sur la gauche. Il faut choisir ensuite *Composants*. Dans la liste qui va apparaître, c'est l'outil *Microsoft Direct Speech Recognition* qu'il faut cocher avant de ressortir.

Un nouvel objet est maintenant disponible dans la barre d'outils : une oreille. C'est l'objet qui nous concerne, son nom par défaut est "DirectSR1".

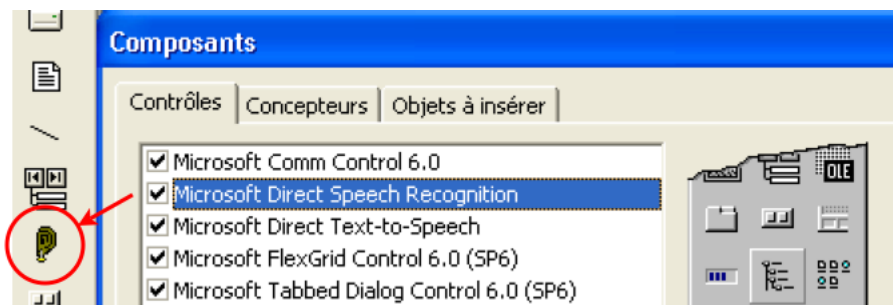


FIG. 4.8: Composant de reconnaissance vocale

4.2.2 Déclaration du dictionnaire

Ce que Microsoft appelle le dictionnaire est en fait la liste des mots reconnaissables. Certaines informations doivent être définies afin que le module de reconnaissance sache qu'il a à faire à un dictionnaire, et à partir d'où se situent les mots à reconnaître.

```
' La variable totaldata représente la liste, on l'initialise ici avec la syntaxe
' de base obligatoire
'-----
totaldata = "[Grammar]" & vbCrLf & _
"langid = 1033" & vbCrLf & _
"type=cfg" & vbCrLf & _
"<Start>"

' Initialisation de la reconnaissance
'-----
Infos.DirSR1.Deactivate                '== On désactive =='
Infos.DirSR1.GrammarFromString (totaldata) '== On charge la variable =='
Infos.DirSR1.Activate                  '== On active =='
```

FIG. 4.9: Code permettant de déclarer le dictionnaire et de l'enregistrer

Chaque mot à reconnaître est à rajouter après une instruction du type : [<Start>] sous la forme d'une chaîne de caractères.

Il y a deux méthodes pour que le dictionnaire et ses paramètres soient enregistrés :

- La première consiste à écrire le tout dans un fichier texte basique type ".txt" et de venir récupérer ce fichier depuis Visual Basic.
- La seconde est d'assembler toutes les commandes et tous les mots dans une variable de type string. Il est également possible de remplir cette variable depuis une base de données en parcourant les entrées de celle-ci.

La seconde méthode offre l'avantage d'un mode de lecture et d'écriture facilité. Cela va permettre à l'utilisateur d'ajouter un nouveau mot reconnaissable sans redémarrer le programme.

Dans le cadre de mon application, la méthode d'ajout et de suppression des mots a surtout été utilisée pour le développement. Le seul cas où l'utilisateur a vraiment la possibilité d'ajouter un mot et de le rattacher à une action est lorsqu'il ajoute une nouvelle couleur et lui attribue un mot d'ordre.

4.2.3 Reconnaissance des mots

En double-cliquant sur l'objet "DirectSR1", une méthode est créée. Cette méthode est appelée à chaque fois qu'un mot du dictionnaire est entendu par le micro. Ensuite, c'est à l'aide de la fonction *SELECT CASE* que le mot entendu est comparé à notre dictionnaire et qu'une action adéquate peut être exécutée.

```

'-----'
'  Cette procédure est appelée chaque fois qu'un mot est détecté dans le micro  '
'-----'
Public Sub DirSR1_PhraseFinish(ByVal flags As Long, ByVal beginhi As Long, _
    ByVal beginlo As Long, ByVal endhi As Long, ByVal endlo As Long, _
    ByVal Phrase As String, ByVal parsed As String, ByVal results As Long)

'  Selon le mot prononcé une fonction est appelée
'-----'
Select Case Phrase

    Case "Command"                                '== Mot de sécurité 1 =='|
        Param.Give

```

FIG. 4.10: Déclaration de la fonction de reconnaissance vocale et début de fonction *SELECT CASE*

4.3 Synthèse vocale

Microsoft propose également dans son kit de développement, "speechSDK51.exe", un moteur de synthèse vocale. La voix de base s'appelle "Microsoft Sam". Elle a une prononciation anglaise et un effet très "robot". Deux autres voix sont incluses dans le kit de développement : Mike et Mary, qui sont légèrement plus fluides. Attention, il faut tout de même installer le moteur de synthèse vocale. Plusieurs moteurs sont disponibles sur le site de Microsoft¹.

Des voix dans d'autres langues sont disponibles sur internet. Certaines sont développées par des sociétés spécialisées et sont donc payantes. Elles sont généralement plus propres et gèrent mieux la prononciation et la ponctuation. Mais les voix disponibles sont exploitables.

Sur le site de Microsoft, il est possible de trouver un moteur de voix française. Ce moteur fonctionne bien avec Visual Basic. Seulement, comme la reconnaissance vocale est en anglais, j'ai trouvé déplacé de parler à la machine en anglais et qu'elle me réponde en français. C'est donc le moteur anglais qui a été retenu.

¹<http://www.microsoft.com/msagent/downloads/user.aspx>

4.3.1 Ajout du composant

Pour ajouter un objet de synthèse vocale dans Visual Basic, il faut faire un clic droit dans la barre d'outils sur la gauche. Il faut choisir ensuite "Composants". Dans la liste qui va apparaître c'est l'outil *Microsoft Direct Text-to-Speech* qu'il faut cocher avant de ressortir.

Un nouvel objet est maintenant disponible dans la barre d'outils : une bouche. C'est l'objet qui nous concerne, son nom par défaut est "DirectSS1".

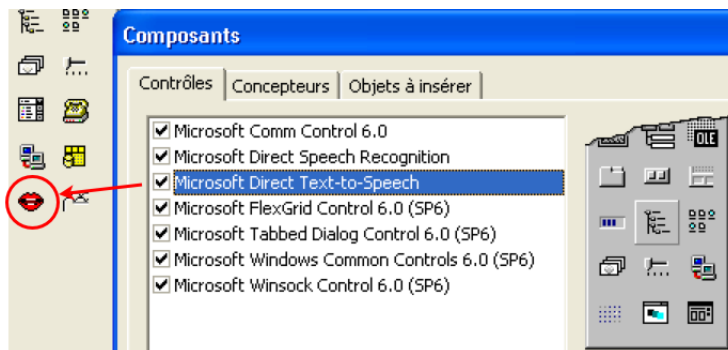


FIG. 4.11: Composant de synthèse vocale

4.3.2 Utilisation de l'objet

Afin de faire parler cet objet, une simple commande tenant sur une seule ligne est utilisée :

```
DirectSS1.Speak ("This is a voice test")
```

4.4 Base de données

Afin de mémoriser différentes informations et ne pas les perdre lorsque le programme est arrêté, il a fallu trouver un moyen de conserver les données dans un fichier externe. La solution la plus adéquate est d'utiliser une base de données.

Le programme Visual Basic étant un produit Microsoft, le choix s'est porté sur le système de base de données ACCESS. La gestion d'une base de données ACCESS au travers de Visual Basic est assez accessible. Des fonctions et des objets sont déjà prévus pour communiquer.

Il aurait été possible de communiquer avec cette base de données à l'aide de commandes SQL. Cette méthode aurait pu me permettre d'utiliser un autre programme de gestion de base de données. Seulement, mon objectif étant d'avoir un résultat efficace rapidement, j'ai opté pour une solution plus abordable pour moi.

4.4.1 Déclaration des objets

Il y a deux types d'objets qu'il est nécessaire de déclarer. Le premier concerne la base de données elle-même et il est du type *dataBase*. Cet objet représente le fichier de la base de données. Le second concerne les tables contenues dans la base de données. Ce deuxième objet est du type *Recordset*. Ils se déclarent comme des variables standards :

```
Public base_donnee as dataBase  
Public table1 as Recordset
```

Il faut ensuite définir ces objets en leur attribuant un fichier donné ou une table précise :

```
Set base_donnee = OpenDatabase (App.Path & "\basededonnee.mdb")  
Set table1 = base_donnee.OpenRecordset ("Table")
```

4.4.2 Parcours d'une base de données

Pour travailler avec une base de données, il est nécessaire de pouvoir naviguer dans les entrées de celle-ci. Plusieurs commandes sont nécessaires dont voici les plus importantes :

table1.MoveFirst : Elle permet de se déplacer sur le premier enregistrement d'une table. Ceci est très utile s'il faut parcourir l'ensemble de la table à la recherche d'une entrée.

table1.MoveNext : Elle permet de se déplacer sur l'entrée suivante. A nouveau utilisé lors de la recherche d'une entrée.

table1.Fields("nom") : C'est une fonction qui permet d'accéder au contenu de la colonne nommée *nom* de l'entrée qui a été sélectionnée plus tôt.

Ensuite, afin de parcourir toutes les entrées jusqu'à la dernière utilisée, il faut utiliser une boucle *Do While Loop*. La condition pour sortir de cette boucle est : `Not table1.EOF`. Cette variable *EOF* est vraie si l'entrée sélectionnée est la dernière de la table.

Lorsque la valeur souhaitée est égale à la valeur de l'entrée, la boucle est interrompue par la commande *Exit Do*. Ainsi pour parcourir une table en testant une entrée, il faut écrire le code suivant :

```
table1.MoveFirst
Do While Not table1.EOF

    If table1.Fields("nom") = test Then Exit Do

    table1.MoveNext
Loop
```

FIG. 4.12: Boucle de parcours de base de donnée

4.4.3 Ajout d'une nouvelle entrée

L'ajout d'une entrée dans une table ne nécessite pas son parcours. En effet, à l'aide de la commande *table1.AddNew* , une entrée est automatiquement ajoutée à la fin. Une fois l'entrée ajoutée, il faut compléter chaque champ l'un après l'autre à l'aide de la commande *table1.Fields* .

Un autre point important lorsqu'une nouvelle entrée est ajoutée, il est nécessaire de mettre à jour la table et de la fermer ensuite. Les commandes pour exécuter ces deux actions sont :

table1.Update : Afin de mettre à jour la table.

table1.Close : Afin de fermer l'accès à la table.

4.4.4 Suppression d'une entrée

Une entrée qui ne serait plus utilisée peut-être supprimée de la base de donnée avec la commande *table1.Delete*.

Avant d'exécuter cette commande, il faut utiliser la boucle de parcours de table afin de sélectionner la bonne entrée à effacer.

Si le numéro de l'entrée est connu, il est également possible d'y accéder directement. Par exemple, dans le cas où les entrées d'une table sont affichées dans une liste, la ligne de l'entrée qui doit être effacée correspond au numéro de l'entrée. Ainsi avec la commande *table1.Move (num_ligne)*, il est possible de pointer sur la bonne entrée et ensuite de l'effacer.

4.5 Communication

Afin de communiquer avec l'ordinateur qui commande le robot Delta, il faut utiliser le protocole TCP/IP afin de passer les transmissions par le réseau de l'école.

Ce protocole regroupe l'ensemble des règles de communication. Il est basé sur une transmission de données selon les adresses IP. Ainsi, chaque machine ayant une adresse IP est capable de communiquer via ce protocole.

4.5.1 Ajout du composant

L'utilisation de ce protocole dans Visual Basic se fait au travers d'un objet appelé *Winsock*. Pour ajouter cet objet dans Visual Basic, il faut faire un clic droit dans la barre d'outils sur la gauche. Il faut choisir ensuite "Composants". Dans la liste qui va apparaître, c'est l'outil *Microsoft Winsock Control 6.0* qu'il faut cocher avant de ressortir.

Un nouvel objet est maintenant disponible dans la barre d'outils. C'est l'objet qui nous concerne, son nom par défaut est "Winsock1".

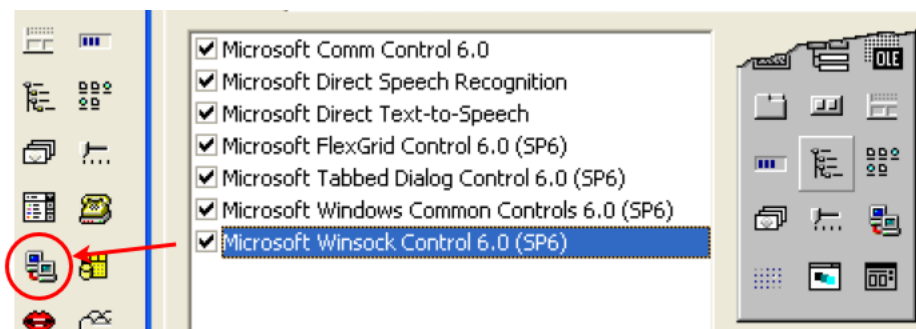


FIG. 4.13: Composant de communication TCP/IP

4.5.2 Gestion de la communication

Le robot est obligé d'avoir la position de Client dans la communication. Il a donc fallu établir que le composant *Winsock* soit le serveur. Afin de gérer la communication, plusieurs commandes sont nécessaires :

Winsock.Close : Elle permet de fermer la communication.

Winsock.Accept requestID : Elle accepte la connexion selon le numéro d'identification qui fait la demande.

Winsock.Listen : Elle met le port ouvert en mode écoute, prêt à recevoir des instructions.

Winsock.SendData(data) : Elle envoie la chaîne de caractères *data*.

Winsock.GetData data : Elle met dans la chaîne de caractères *data* les données reçues.

Winsock.State : Elle retourne le code d'état de la connexion.

4.5.3 Réception de données

Un événement existe lorsque l'objet reçoit des données. Une fonction *Select Case* est alors utilisée pour trier les données et agir en conséquence. Voici comment est déclaré l'événement :

```
Public Sub Winsock1_DataArrival (ByVal bytesTotal as Long)
```

C'est à l'intérieur de cet événement que la commande *winsock1.GetData* est utilisée.

4.5.4 Envoi de données

L'envoi de données peut se faire à tout moment. Seulement, pour que cela fonctionne, il faut que le Client attende des données et qu'il écoute régulièrement. La commande pour envoyer une chaîne de caractère est simplement :

```
winsock1.SendData ("message a envoyé")
```

4.5.5 Protocole personnel

Afin de communiquer avec le robot, j'ai établi un protocole simple permettant de structurer mon programme. Il est composé d'un nombre à trois chiffres et chaque nombre a une fonction très précise dans la programmation. Ainsi, un envoi ne comportant pas le bon code sera négligé.

Dans certains cas, des données supplémentaires sont demandées, comme des coordonnées par exemple. Elles sont ajoutées à la suite du code à trois chiffres.

- 111** : Signal avertissant que la transmission est faite et que le robot est prêt

- 200** : Sélection du mode automatique
- 210x** : Nombre de boîtes à produire, x étant ce nombre
- 220** : Mode automatique : chocolat suivant
- 230cxy** : Mode automatique : la coordonnée de prise est c et les coordonnées de pose sont x et y
- 240** : Mode automatique : éjection des chocolats présents

- 300** : Sélection du mode pas-à-pas
- 320x** : Mode pas-à-pas : envoi de la coordonnée de prise, x est la coordonnée de prise du chocolat
- 330xy** : Mode pas-à-pas : envoi des coordonnées de pose, x et y sont les coordonnées de pose pour la boîte
- 340** : Mode pas-à-pas : éjection des chocolats présents
- 350** : Mode pas-à-pas : signal avertissant que la boîte est pleine
- 360** : Mode pas-à-pas : autorisation au robot de fermer la boîte et de l'amener à la sortie

- 500** : Signal avertissant que le processus est terminé
- 600** : Signal avertissant que la boîte est prête à être remplie
- 800** : Signal avertissant que la boîte a été éjectée du réservoir
- 999** : Information permettant au robot de quitter son programme

CHAPITRE 5

Présentation de l'interface

5.1 Introduction

L'interface a été complètement développée en Visual Basic. Ce programme a l'avantage de posséder tous les outils pour développer les fonctions nécessaires à la réalisation du travail de diplôme.

Toute l'interface a été étudiée afin qu'un utilisateur non averti puisse l'utiliser sans contrainte. Les boutons doivent comporter des explications claires, mais l'utilisateur ne doit pas être confronté à de grosses quantités de texte.

5.2 Présentation de l'aspect visuel

Les modes principaux que sont *Automatique*, *Pas-à-pas* et *Nouveau Produit* se ressemblent d'un point de vue de la conception. Ils sont faits afin que l'utilisateur soit à l'aise.

5.2.1 Aspect général

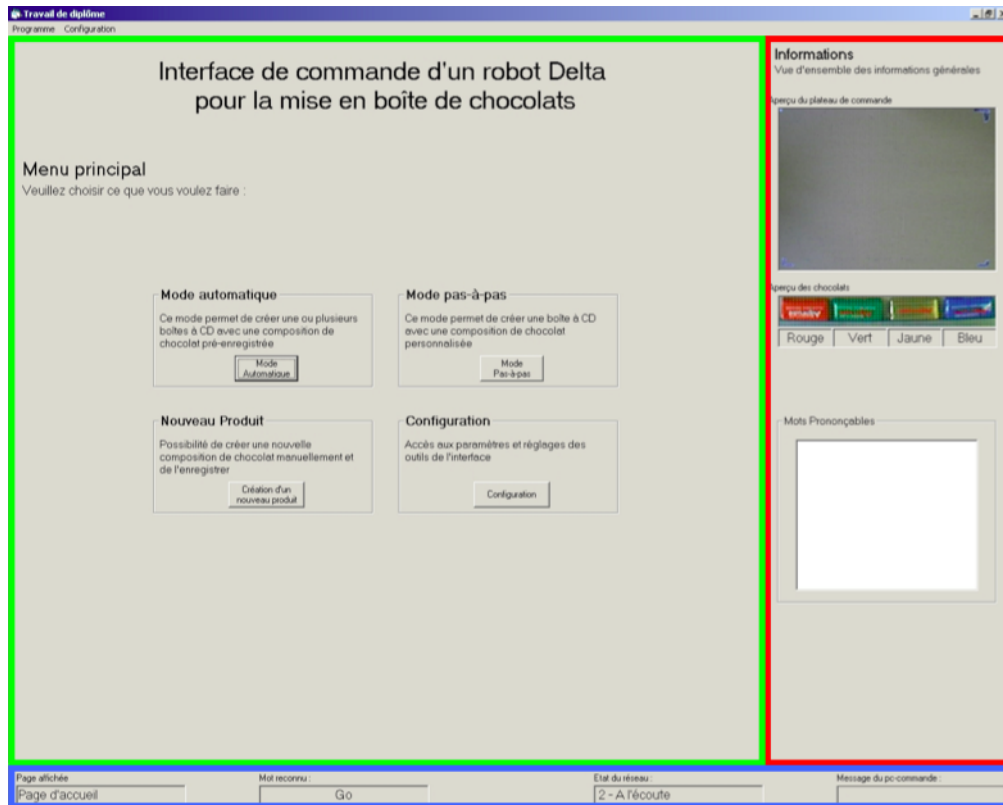


FIG. 5.1: Présentation générale de l'interface

L'interface est composée de trois zones principales :

- Zone verte : C'est la zone de travail. C'est dans cette partie que l'utilisateur va pouvoir interagir. Les différents menus s'afficheront dans cette zone.
- Zone rouge : C'est la zone de visualisation. Les flux vidéo des webcams sont affichés dans cette partie ainsi que des informations propres à la page affichée dans la zone de travail.
- Zone bleue : Barre d'informations. Cette barre d'information permet de connaître quelques informations générales à tout moment dans le programme.

A noter également que l'interface est munie de menus déroulants offrant quelques accès supplémentaires aux menus.

Zone de travail

Chaque page de travail est présentée par son titre et une courte description de sa fonction. Les dimensions de la page s'adaptent à la résolution de l'écran utilisé. Ainsi la zone de travail couvre les trois quarts de la largeur de l'écran.

C'est cette zone qui va varier le plus durant l'utilisation de l'interface. Elle reçoit le menu principal, les menus des différents modes ainsi que le menu configuration qui contient les paramètres de tous les outils. Ces différentes fenêtres seront expliquées plus bas dans ce chapitre.

Zone de visualisation

Il est possible de voir la zone de visualisation en trois parties. La première est composée de l'affichage des vidéos. La première vidéo correspond à la zone de commande et la seconde vidéo affiche les chocolats prêts à être pris. La deuxième zone informe l'utilisateur des mots qu'il peut prononcer dans l'étape du programme dans laquelle il se trouve. En effet, comme expliqué au chapitre 4.2.1 sur l'analyse des risques de la reconnaissance vocale, un des bons moyens de protéger la mauvaise compréhension d'un mot est d'autoriser son écoute qu'à des moments particuliers. Ainsi l'utilisateur sait quels mots seront compris ou non, cela lui facilitera l'utilisation du programme.

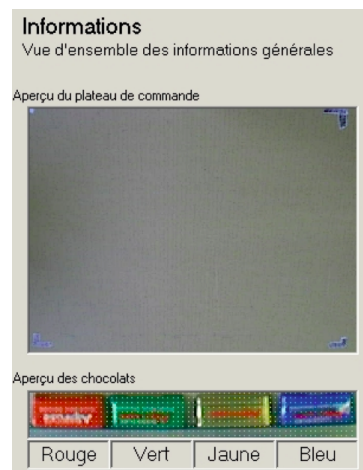


FIG. 5.2: Partie supérieure de la zone de visualisation

La dernière zone n'est pas toujours visible. Elle contient de petites informations liées à la page de la zone de travail activée. Par exemple, en mode pas-à-pas, la couleur sélectionnée est affichée.

Zone d'informations

La barre d'informations est composée de quatre messages. Le premier informe l'utilisateur de la page affichée dans la zone de travail. Le deuxième signale le mot qui a été reconnu par le moteur de reconnaissance vocale. Cela permet ainsi de contrôler si l'ordre donné par l'utilisateur a été bien compris. Le troisième concerne la communication du réseau. Il affiche la situation de la communication avec l'ordinateur de commande. Et pour finir le quatrième message indique les informations ou les ordres donnés par l'ordinateur de commande.



FIG. 5.3: Barre d'informations de l'interface

5.2.2 Menu principal

La page d'accueil est le centre du programme. C'est à partir d'elle que toutes les autres fenêtres sont accessibles. Quatre boutons sont donc mis en place sur la page avec une courte description de leur fonction.

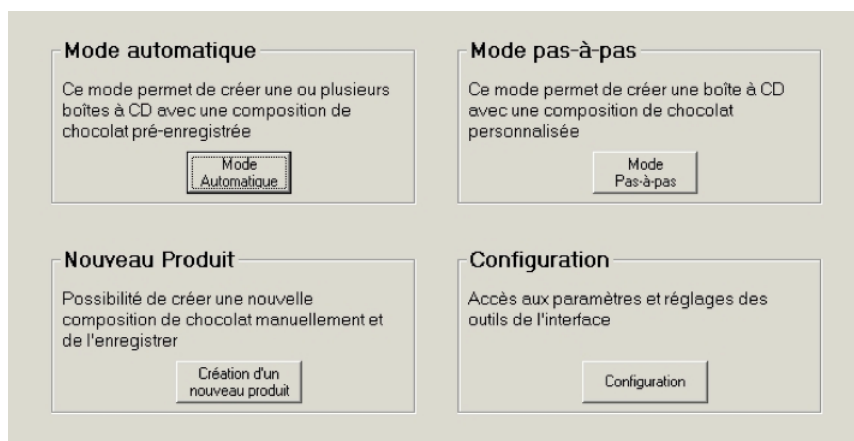


FIG. 5.4: Boutons de la page d'accueil et leur description

Toutes les autres pages ont un bouton situé en haut à droite qui permet de revenir à cette page et d'annuler l'action en cours.

5.2.3 Fenêtre du mode automatique

Le mode automatique permet de produire une composition pré-enregistrée. Afin de guider l'utilisateur dans la création de son produit, des étapes sont clairement définies.

La première étape consiste à choisir la composition que l'utilisateur veut créer. Il peut le faire à l'aide de la liste déroulante comprenant les compositions enregistrées. Lorsque c'est fait, la composition est affichée de manière graphique dans le tableau central.

Ensuite, l'utilisateur doit entrer le nombre de fois qu'il veut que la composition choisie soit créée. Une fois validé, il est possible de lancer la production.

La troisième et dernière partie où l'utilisateur intervient consiste à lancer la production en validant à l'aide du pointeur lumineux et de l'ordre vocal *Ok*.

The screenshot displays a software interface for creating chocolate compositions. It is divided into four main sections:

- 1.- Choisir une composition**: Features a dropdown menu labeled 'Liste de choix:' with 'Milde' selected. Below it, a 3x4 grid of colored squares represents the composition. A legend on the right identifies the colors: 0. Vide (white), 1. Rouge: Gianduja, 2. Vert: Noisettes, 3. Bleu: Lait, 4. Jaune: Noir.
- 2.- Sélection du nombre de boîte à produire**: Includes a text prompt 'Veuillez choisir le nombre de boîte que vous voulez créer, entre 1 et 5 : (Entrée pour valider)' and a small input field.
- 3.- Validation de la commande**: Contains a text prompt 'Pour lancer la production, il faut valider la commande à l'aide du curseur lumineux. Pour cela, le curseur doit être positionné dans la zone de validation et dites "Ok".' and a small input field.
- 4.- Etat de la commande**: Shows a progress bar with five segments, each labeled with a step: '1.- Choix', '2.- Nombre de boîte', '3.- Validation', 'Production', and 'Terminé'. The first three segments are filled with blue bars, indicating the current progress.

FIG. 5.5: Eléments de la page de composition automatique

La quatrième étape n'en est pas vraiment une. En réalité, c'est une indication de l'avancement de la production. Ainsi l'utilisateur peut connaître où il se trouve dans la création de son produit.

5.2.4 Fenêtre du mode pas-à-pas

Le mode pas-à-pas permet de créer une composition étape par étape par l'utilisateur. Il faut donc faire en sorte qu'il soit bien aiguillé.

Pour créer un produit, il faut exécuter trois étapes. En premier, il faut que le système amène une nouvelle boîte afin de la remplir. D'une fois que c'est fait, l'utilisateur peut remplir manuellement la boîte chocolat par chocolat.

Lorsque la boîte est remplie, un message apparaît afin d'avertir l'utilisateur. Il peut donc dès lors valider la création de la boîte à l'aide du bouton prévu à cet effet ce qui fermera la boîte et l'évacuera.

Ces étapes doivent être simples d'utilisation et quelques explications permettent de guider l'utilisateur.

The image shows a simplified representation of a step-by-step mode window. It is divided into three main sections:

- Lancer la création d'une boîte**: This section contains instructions: "Pour lancer la création d'une nouvelle boîte personnalisée, mettez le curseur lumineux dans la zone de sélection et dites : 'New Creation'. Sinon, cliquez sur le bouton ci-dessous. La barre de progression indique le moment où la boîte est arrivée sur la zone de travail." Below the text is a button labeled "Nouvelle boîte" and a progress bar with labels "Ejection", "Convoyage", and "Pose de la boîte".
- Aperçu de la composition**: This section features a legend on the left with the following items:
 - 0. Vide
 - 1. Rouge : Gianduja
 - 2. Vert : Noisettes
 - 3. Bleu : Lait
 - 4. Jaune : NoirTo the right of the legend is a 4x4 grid representing the chocolate box composition.
- Terminer la création de la boîte**: This section contains instructions: "Lorsque la boîte est pleine, un message s'affiche ci-dessous. Afin de fermer la boîte et de l'amener à la sortie, mettez le curseur lumineux dans la zone de validation et dites : 'Ok', ou alors cliquez avec la souris sur le bouton ci-dessous." Below the text is a button labeled "Terminer la création".

FIG. 5.6: Représentation simplifiée de la page du mode pas-à-pas

5.2.5 Fenêtre du mode d'enregistrement de produit

Pour enregistrer un nouveau produit, les étapes doivent être respectées afin que tout se passe correctement. Encore une fois, il est important que l'utilisateur n'ait pas besoin de se creuser la tête pour agir correctement. Ainsi les démarches ont été définies en quatre étapes.

La première chose à faire est de donner un nom au produit qui est en train d'être créé. Ensuite, il faut remplir *virtuellement* la boîte. Pour cette étape, la méthode est la même que pour la méthode pas-à-pas sans le contrôle du robot, donc la visualisation graphique est la même.

D'une fois que la boîte est pleine, il est possible de passer à l'étape suivante qui est de donner une description au produit créé. Cette description sera reprise lorsque l'utilisateur choisira ce produit dans le mode automatique.

La dernière étape est bien sûr l'enregistrement du produit dans la base de données. L'utilisateur n'a plus qu'à cliquer sur un bouton prévu pour cette fonction.

The image shows a simplified representation of a product creation interface, divided into four numbered steps:

- 1.- Nom du produit**: A section with the instruction "Entrez ici le nom de votre nouveau produit à enregistrer :" followed by a text input field labeled "Nom".
- 2.- Aperçu de la composition**: A section containing a legend box on the left and a 3x3 grid on the right. The legend lists: 0. Vide, 1. Rouge : Gianduja, 2. Vert : Noisettes, 3. Bleu : Lait, 4. Jaune : Noir.
- 3.- Description**: A section with the instruction "Donnez une brève description de votre composition :" followed by a long text input field.
- 4.- Enregistrer le produit**: A section with instructions: "Pour enregistrer votre composition, il faut que la boîte soit pleine. Ensuite, pour valider l'enregistrement soit vous positionnez le curseur lumineux sur la zone de validation et dites 'Ok', soit vous cliquez sur le bouton ci-dessous." Below the text is a button labeled "Enregistrer".

FIG. 5.7: Représentation simplifiée de la page du mode de création d'un nouveau produit

5.3 Paramètres de configuration

C'est dans le menu *Configuration* que l'utilisateur a accès à tous les paramètres des outils. Il est important d'offrir une certaine souplesse à ces paramètres, afin que l'utilisateur puisse les adapter à son environnement.

5.3.1 Aspect du menu

Les différents outils du menu configuration sont accessibles par un système d'onglets. Ce système est propre et permet d'accéder facilement à chaque outil. L'utilisateur pourra ainsi se retrouver facilement lorsqu'il doit gérer un des outils.

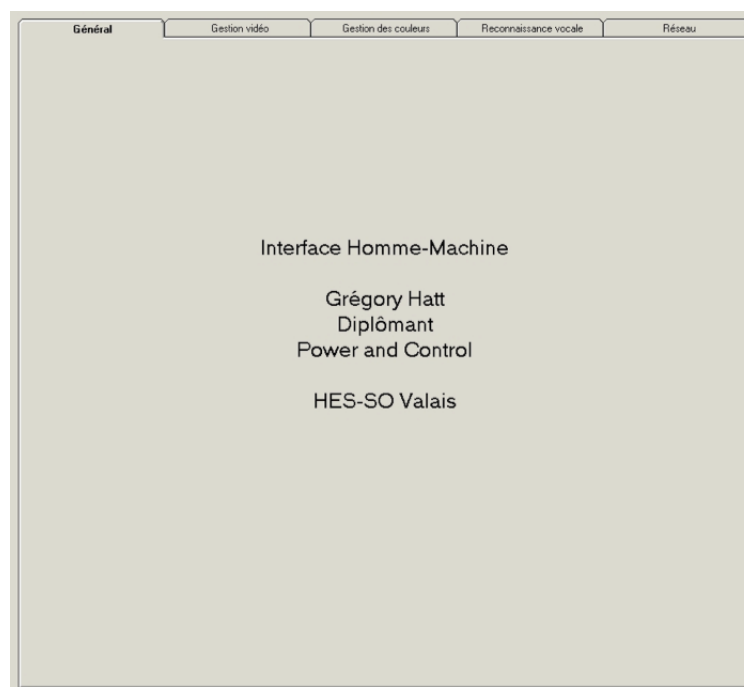


FIG. 5.8: Menu configuration : Général

L'onglet *Général* fournit des informations sur la réalisation du programme.

5.3.2 Gestion Vidéo

La gestion de la vidéo étant la partie la plus complexe, les paramètres de réglages sont nombreux. Voici tout d'abord la présentation générale du menu :

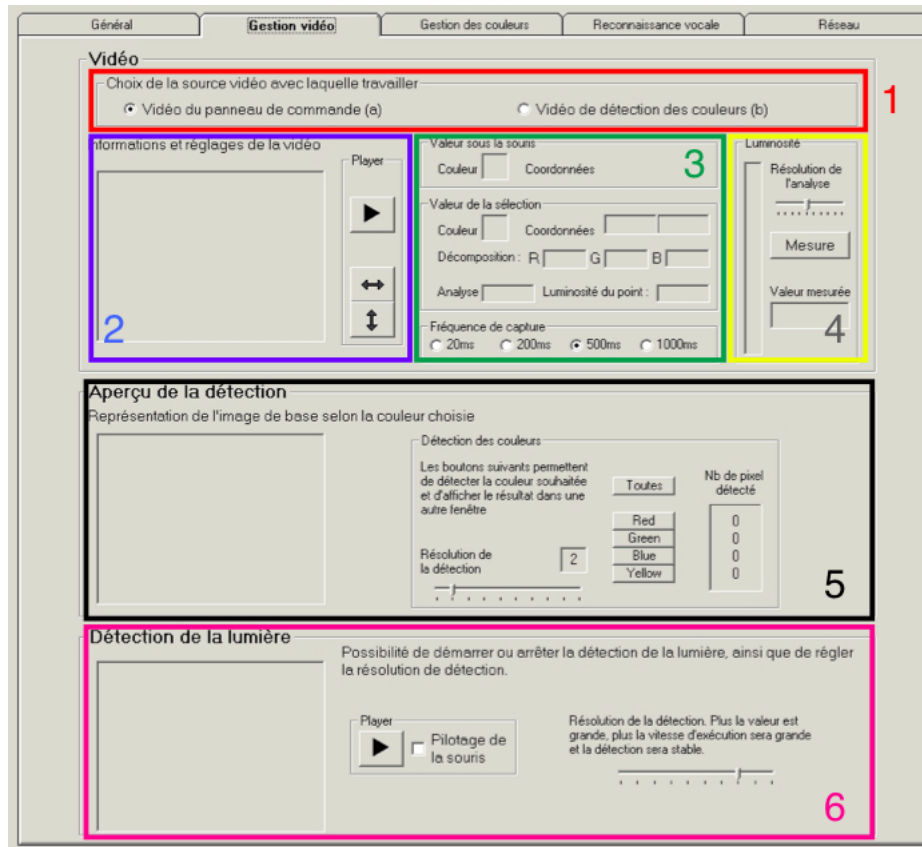


FIG. 5.9: Menu configuration : Gestion de la vidéo

- 1 L'utilisateur peut choisir quelle webcam est analysée : la vidéo de commande ou la détection des chocolats.
- 2 La capture d'image de la source choisie est affichée dans cette fenêtre, ainsi que les boutons de contrôles.
- 3 Les paramètres de compositions des couleurs sont visibles dans cette zone.
- 4 La luminosité est analysée selon la résolution voulue.
- 5 Chaque couleur est analysée séparément. Le résultat de l'analyse est affichée dans la fenêtre de cette zone.
- 6 Dans cette zone, un aperçu de la détection du pointeur lumineux est affiché dans la fenêtre sur la gauche.

La zone 1 sert uniquement à choisir la webcam à analyser. Elle ne sera donc pas plus détaillée. Par contre les autres zones ont besoin de plus d'explications.

Zone 2

Le bouton *Play* sert à démarrer la capture de l'image dans la fenêtre à côté. Lorsqu'il est activé, il est remplacé par un bouton *Pause* et un bouton *Ampoule*. Le premier sert naturellement à mettre la capture en pause, alors que le second active l'augmentation de la luminosité. Attention, dans ce cas là, la capture est mise en pause.

Les deux autres boutons permettent d'effectuer un effet miroir à l'image. Cette option permet de réagir en fonction de la position de webcam, afin de faire en sorte que l'image soit dans le bon sens par rapport à l'utilisateur. Le sens des flèches indiquent le sens de basculement de l'effet miroir.

Zone 3

Dans cette zone, plusieurs outils sont présents. Le premier correspond à la couleur qui se trouve sous la souris et ses coordonnées en *Live*. Le deuxième se met à jour à chaque clic de souris sur l'image. Cet outil reprend les mêmes paramètres que le premier outil, mais en plus, il décompose les paramètres RVB, donne la luminosité du point et analyse si la couleur est reconnue. Le dernier outil permet de changer la fréquence de capture.

The image shows a software interface for video analysis parameters, organized into three distinct sections with a light gray background and rounded corners.

- Valeur sous la souris**: This top section contains two input fields: "Couleur" (with a small color selection icon) and "Coordonnées".
- Valeur de la sélection**: The middle section contains several input fields: "Couleur" (with a color icon), "Coordonnées" (with two adjacent input boxes), "Décomposition : R" (with an input box), "G" (with an input box), "B" (with an input box), "Analyse" (with an input box), and "Luminosité du point :" (with an input box).
- Fréquence de capture**: The bottom section features four radio buttons for selecting the capture frequency: "20ms", "200ms", "500ms" (which is currently selected), and "1000ms".

FIG. 5.10: Paramètres de l'analyse de la vidéo

Zone 4

La luminosité de l'image est analysée dans cette zone et sur demande. Afin d'accélérer l'analyse, l'utilisateur a le choix de la résolution. La valeur du résultat est donnée entre 0 et 255 ou alors sur la jauge en pourcent.

Zone 5

Dans cette zone, la détection de la couleur est accessible à l'utilisateur. Il y a 5 boutons : 4 pour détecter les couleurs de manière séparée et 1 pour les détecter toutes. Lorsqu'un bouton est pressé, la couleur choisie est affichée dans la fenêtre à gauche et selon la résolution définie en dessous des boutons.

Afin de pouvoir comparer les différents résultats selon les paramètres de détection, le nombre de pixels détectés pour chaque couleur est affiché.

Zone 6

Lorsque la détection du point lumineux est activée, il est possible de connaître quels points répondent aux critères de détection. Ces points s'affichent en noir dans la fenêtre sur la gauche de la zone.

Le bouton *Play* active la détection et il se change en bouton *Pause* lorsqu'il est pressé. Une coche permet d'enclencher le pilotage de la souris. Lorsqu'il est activé, l'endroit où le point lumineux est détecté donne les coordonnées de la souris. Si l'utilisateur dit *Select*, cela génère un clic gauche de souris.

La résolution de la détection peut être modifiée. Attention, si la détection est trop grossière, le point lumineux ne pourra pas être détecté. Par contre si elle est trop fine, le temps de détection sera trop long et rendra la détection inutilisable. Il faut donc trouver le meilleur compromis.

5.3.3 Gestion des couleurs

Le menu de gestion des couleurs permet d'enregistrer de nouvelles couleurs ou de les supprimer. Cela concerne les zones 1 et 2. Dans la zone 3, nous retrouvons les réglages de détection par différences entre les paramètres RVB.

Création d'une nouvelle couleur secondaire

1. - Pour commencer l'apprentissage d'une nouvelle couleur cliquez sur le bouton "Go".

2. - Ensuite sélectionnez sur l'image de la webcam dans le formulaire (a), la couleur que vous souhaitez apprendre.

3. - Voici les paramètres de la couleur sélectionnées ainsi qu'un échantillon :

4. - Veuillez régler les paramètres de tolérance qui vous conviennent pour cette couleur :

5. - Veuillez nommer votre nouvelle couleur ainsi qu'un nom anglais qui sera utilisé pour la reconnaissance vocale :

6. - Résumé avant

Suppression de couleurs

Sélectionnez une ligne et cliquez sur supprimer pour effacer l'enregistrement d'une couleur

Nom	Ordre	Valeur R	Valeur B	Valeur G	Tol. supérieure	Tol. inférieure	Luminosité
Rouge	Rouge	151	9	0	75	70	53
Cuivre	Copper	178	172	105	30	24	152

Paramètres de détection des couleurs principales

Voici les paramètres permettant de détecter les couleurs. Pour qu'une couleur soit détectée en tant que telle, il faut que les conditions S et T soient vraies.

Valeur S

Valeur T

Enregistrement

Sauver Enregistrer les valeurs

Charger Remettre les valeurs enregistrées

Défaut Remettre les valeurs standards

FIG. 5.11: Menu configuration : Gestion des couleurs

Zone 1 : Création d'une nouvelle couleur

La création d'une nouvelle couleur se fait en six étapes. Il est possible à tout moment d'annuler tout ce qui a été fait à l'aide du bouton annuler.

1. Pour commencer, il faut cliquer sur le bouton *GO* ou donner la commande vocale du même nom. A ce moment là, le programme va passer sur l'affichage de l'onglet de la gestion de vidéo.
2. L'étape suivante consiste à cliquer sur la fenêtre de capture d'image afin de sélectionner une couleur.

3. Dès qu'une couleur est sélectionnée, l'onglet de gestion des couleurs est ré-activé. La couleur choisie et ses paramètres RVB sont affichés. Après trois secondes, un message apparaît demandant si la couleur convient : si oui, nous passons à l'étape suivante, si non, il faut recommencer l'étape 2.
4. Une fois la couleur validée, la méthode de détection des couleurs par proximité des paramètres RVB est utilisée afin de montrer le résultat de la détection sur la fenêtre centrale. Si le résultat ne convient pas, les marges de détection sont modifiables et le bouton *Refresh* est là pour mettre à jour la détection.
5. Dans cette étape, le nom de la couleur est choisi par l'utilisateur ainsi que le mot d'ordre qui lui sera attribué. Attention, le mot d'ordre doit être en anglais afin que la reconnaissance vocale fonctionne. Une fois les noms validés, un résumé est affiché dans l'étape 6.
6. un résumé de la couleur et de ses paramètres est affiché. En cliquant sur *Valider*, un contrôle est fait afin de déterminer si le nom ou le mot d'ordre de la couleur existe déjà. Si cela n'est pas le cas, le bouton *Enregistrer* devient actif. Lorsque se dernier est pressé, la couleur est enregistrée dans la base de donnée.

Zone 2 : Suppression d'une couleur personnelle

Toutes les couleurs personnelles qui sont enregistrées apparaissent dans le tableau de cette zone. Afin de supprimer une couleur, celle-ci doit être sélectionnée dans le tableau avant de cliquer sur le bouton *Supprimer*.

Zone 3 : Réglages de détection de couleur par différences entre les paramètres RVB

Les différences entre les paramètres RVB sont à définir dans cette zone. Pour chaque couleur, les différences doivent être paramétrées.

Comme cela a été vu dans l'explication du mode de détection, deux comparaisons suffisent pour établir si la couleur est la bonne. Je les ai appelées *Valeur S* et *Valeur T*. Il faut que ces deux paramètres soient vrais pour définir à quelle couleur nous avons à faire.

Il est important que ces paramètres puissent être sauvegardés par l'utilisateur, c'est pourquoi il est possible d'enregistrer et de charger ces paramètres depuis la base de données.

5.3.4 Reconnaissance vocale

L'accès à la reconnaissance vocale par l'utilisateur est assez simple. Il a la possibilité d'enregistrer un nouveau mot ou d'en supprimer. Les mots ont la possibilité d'être protégés. Si dans la liste un mot est accompagné d'une croix, cela signifie qu'il ne peut pas être supprimé par le programme. Si ce mot doit vraiment être supprimé, la seule solution est de le faire directement dans le programme *ACCESS*. Il est possible de mettre cette protection sur un nouveau mot en cochant la case prévue à cette effet.

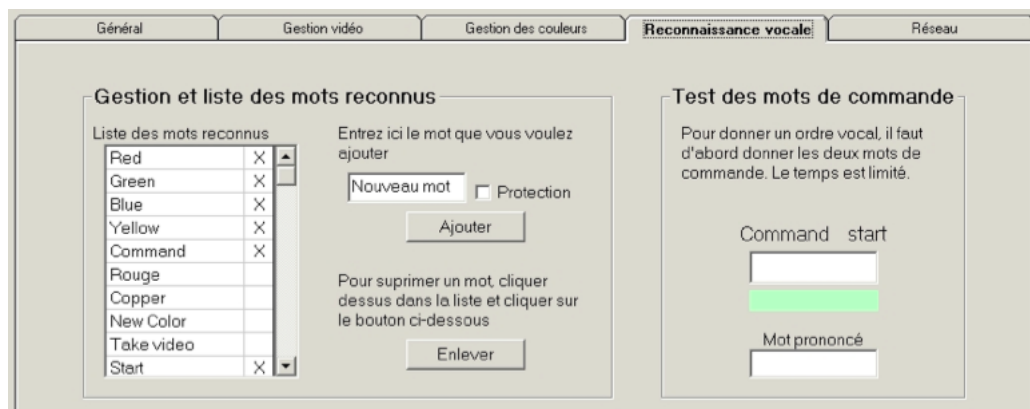


FIG. 5.12: Menu configuration : Reconnaissance vocale

La seconde partie de cet onglet permet de tester la double commande de sécurité. Les mots de commandes s'affichent en rouge lorsqu'ils sont prononcés de manière correcte.

5.3.5 Réseau

Les paramètres du réseaux comme le port utilisé, le nom et l'adresse IP de l'ordinateur ainsi que l'état de la connexion du réseau sont affichés dans la partie de gauche.

Les commandes dans la partie de droite permettent d'agir sur la communication. Il est possible de fermer la connexion, de la mettre sur écoute ou de lire les données reçues. Une fonction permet également d'envoyer manuellement la chaîne de caractères voulue.

The screenshot shows a software interface with a tabbed menu at the top. The tabs are 'Général', 'Gestion vidéo', 'Gestion des couleurs', 'Reconnaissance vocale', and 'Réseau'. The 'Réseau' tab is selected. The main area is divided into two columns. The left column, titled 'Paramètres', contains four input fields: 'N° de port en local' (value: 3000), 'Nom de l'hôte local' (value: we2163), 'N° IP local' (value: 153.109.2.162), and 'Etat de la connexion' (value: 2 - À l'écoute). The right column, titled 'Commandes', contains a text box with the instruction 'Mets la connexion en écoute. Cela permet à l'ordinateur principal de se connecter.', a button 'Ecoute', another text box with the instruction 'Pour fermer la connexion et stopper toutes communications.', a button 'Fermer', a text box labeled 'Données reçues :', a text box labeled 'Envoi de texte: (une fois connecté)', and a button 'Envoyer'. At the bottom, there is a section titled 'Journal des erreurs :' with a large empty text area and a button 'Vider'.

FIG. 5.13: Menu configuration : Réseau

En cas d'erreur dans la connexion, l'erreur est inscrite dans la liste d'erreurs. Cette fonction ne va pas plus loin qu'un simple affichage. Les erreurs ne sont pas conservées.

CHAPITRE 6

Présentation de la réalisation pratique

L'application pratique qui utilise les outils de reconnaissance vocale et d'analyse d'image s'est faite autour du robot Delta. Une application de démonstration avait été réalisée sur ce robot par Christophe Gabbud. Cette application mettait des jetons de couleurs dans une boîte de cd.

J'ai repris la base de son application afin de mettre des chocolats dans les boîtes de cd. Ainsi les bases mécaniques pouvaient être ré-utilisée. Il y a une seule partie mécanique que j'ai du refaire : le réservoir à chocolat¹. Il a fallu en dessiner un nouveau adapté aux chocolats.



La programmation du robot se fait en TurboPascal. Certaines commandes sont spécifiques au robot. Elles sont décrites en détails dans la documentation Demau-rex S.A. Dans le cadre de son travail de diplôme, Christophe Gabbud a écrit un mode d'emploi afin de programmer rapidement une application à l'aide de ces commandes propres au robot.

Ce document étant très bien fait, la programmation s'est bien déroulée. Je ne connaissais que très vaguement le langage de programmation TurboPascal, mais à l'aide de documentations trouvées sur internet et le programme de base, j'ai pu atteindre mes objectifs.

¹La réalisation du réservoir se trouve dans l'annexe A

6.1 Remplissage automatique

Le mode automatique permet à l'utilisateur de remplir rapidement une boîte de douze chocolats selon un modèle enregistré. Pour exécuter cette tâche, le robot doit interroger l'ordinateur d'analyse afin de connaître les coordonnées de prise et de pose du chocolat. Afin de faire communiquer les deux programmes, un protocole de communication a été défini au chapitre 5.5.5.

6.1.1 Principe

Lorsque l'utilisateur choisi dans l'interface le mode automatique, un signal contenant le code *200* est transmis au robot. Celui-ci entre alors dans une séquence réservée à ce mode. Le robot va dès lors se mettre en attente du nombre de boîtes qui doivent être produites.

Une fois que le robot reçoit le code *210*, il récupère le quatrième caractère de la trame qui correspond au nombre de boîte à produire.

Ensuite, le robot va commencer la production de la première boîte. Il éjecte une boîte du réservoir et envoie le code *800* à l'interface. Dès que la boîte est arrivée en position de remplissage, le robot envoie le signal correspondant : *600*.

Tout de suite après, le robot va demander à l'interface de lui communiquer les coordonnées de prise et de pose de chocolats en envoyant le code *220*.

L'interface va analyser les chocolats qui sont disponibles. S'il y en a un qui fait partie de la composition choisie et qui n'est pas déjà pris, alors l'interface va transmettre au robot le signal *230cxy*. Le quatrième caractère correspond à la coordonnée de prise du chocolat et peut aller de 0 à 3 selon les quatre positions possibles de prise. Les caractères *x* et *y* donnent la position dans la boîte où le robot devra poser le chocolat.

Si aucune des couleurs présentes ne sont utiles, l'interface va envoyer un message au robot, *240*, lui demandant de lancer la procédure de vidage des chocolats. Toutes les positions seront vidées et des nouveaux chocolats seront amenés.

Si un chocolat a pu être posé dans la boîte, le compteur de chocolat augmente jusqu'à douze. Une fois la boîte remplie, le robot le signale à l'interface par le code *500*. Si c'est la dernière boîte à produire, le robot sort du mode automatique. Sinon, la production d'une boîte est relancée.

6.1.2 Structure de la programmation

Afin de développer au mieux la transmission entre les deux programmes, j'ai écrit la structure de la logique en parallèle.

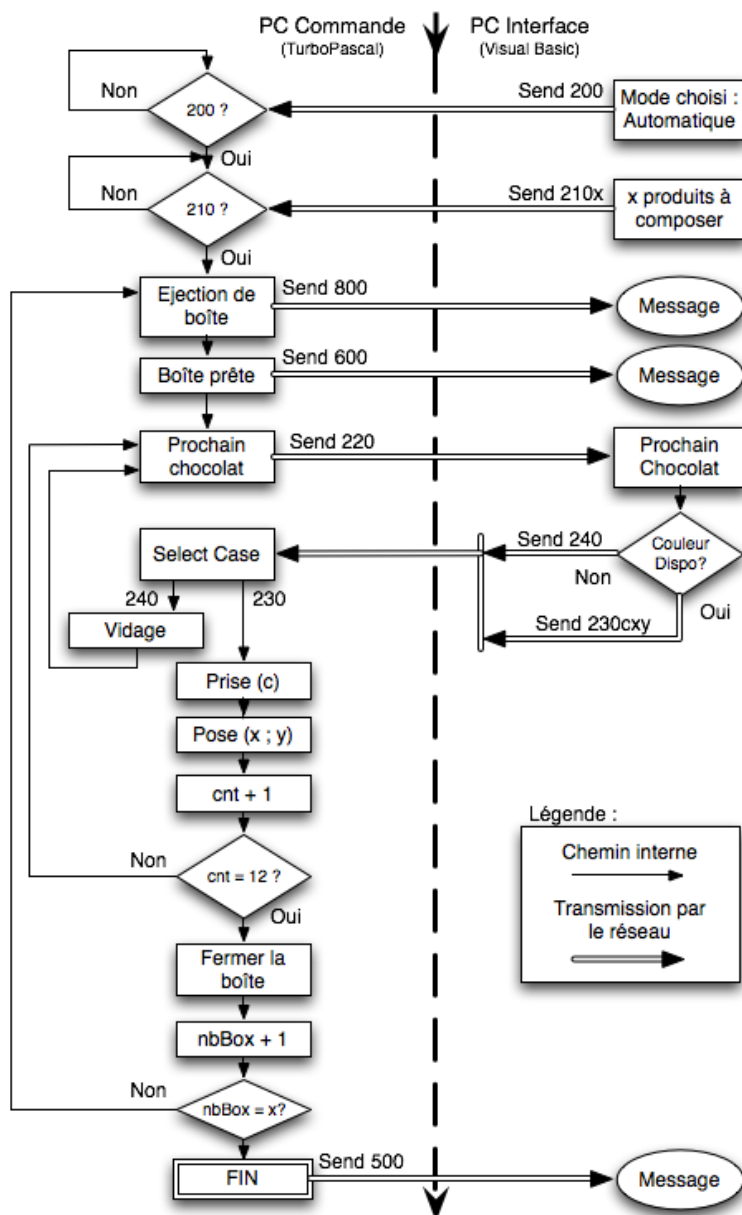


FIG. 6.1: Structure du programme du mode automatique

6.2 Remplissage pas-à-pas

Le mode étape par étape permet de composer une boîte de chocolats selon le désir de l'utilisateur. Le robot va attendre les indications de l'interface avant de faire chaque étape.

6.2.1 Principe

Lorsque l'utilisateur choisi dans l'interface le mode pas-à-pas, un signal contenant le code *300* est transmis au robot. Celui-ci entre alors dans une séquence réservée à ce mode et le robot va commencer la production de la première boîte. Il éjecte une boîte du réservoir et envoie le code *800* à l'interface. Dès que la boîte est arrivée en position de remplissage, le robot envoie le signal correspondant : *600*.

Ensuite, le robot va se mettre en attente d'une coordonnée de prise. L'utilisateur va choisir une couleur à l'aide du pointeur lumineux. Si la couleur est disponible dans les chocolats présents, l'interface va envoyer le signal *320x*, *x* étant la position de prise. Si non, le code *340* est envoyé afin de faire un vidage des chocolats présents.

Dès que le robot reçoit le code *320*, il se déplace à la position *x*, prend le chocolat présent et attend les coordonnées de pose.

Lorsque l'utilisateur choisit une coordonnée de pose, elle est envoyée avec le signal *330xy* si la place n'est pas déjà occupée. Le quatrième caractère correspond à la colonne et le cinquième à la ligne. Le robot pose alors le chocolat pris plus tôt.

Lorsque douze chocolats ont été posés, le robot transmet le code *350* à l'interface pour lui signaler que la boîte est pleine.

Dès lors, l'utilisateur peut confirmer la création de la boîte. A ce moment là, l'interface va envoyer le code *360* au robot pour qu'il ferme la boîte et l'évacue. Une fois que c'est fait, le programme sort du mode pas-à-pas.

6.2.2 Schéma de programmation

Voici la structure de la logique du programme et les communications entre les deux programmes.

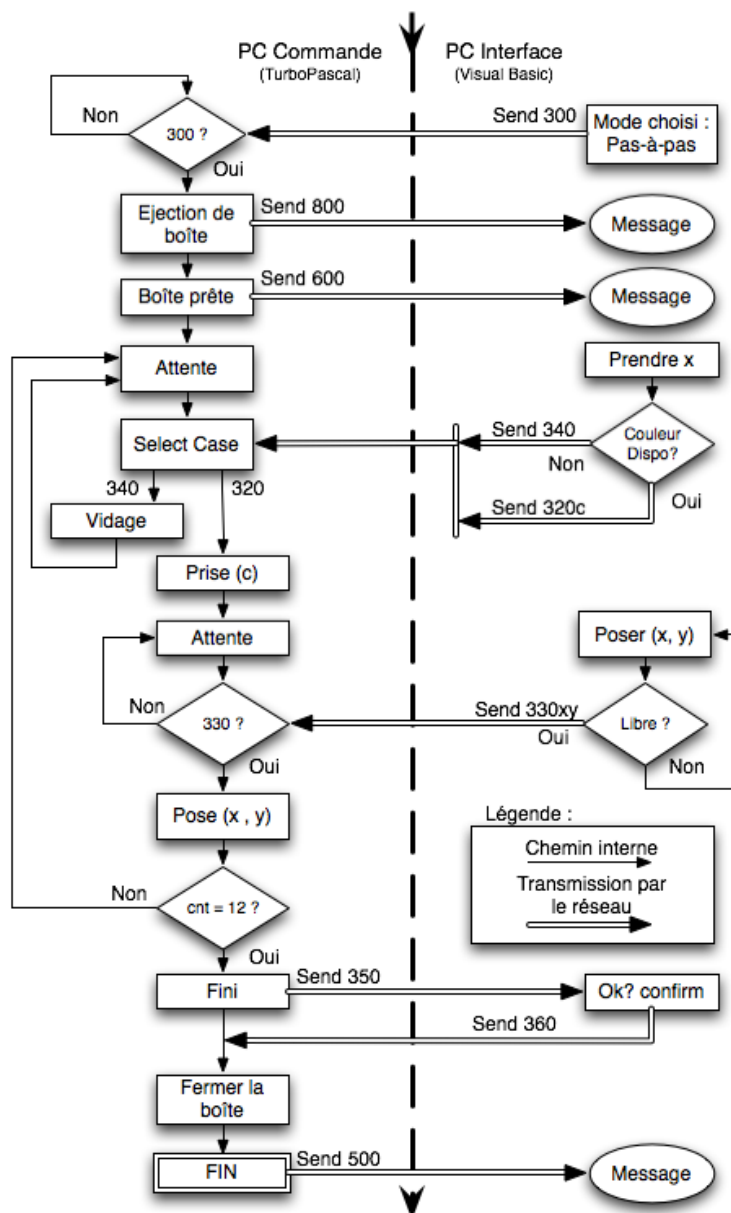


FIG. 6.2: Structure du programme du mode pas-à-pas

CHAPITRE 7

Conclusions

Les interfaces Homme-Machine présentent un réel intérêt pour le futur. Les machines prennent tous les jours une plus grande importance dans nos vies et interagir de manière naturelle avec elles devient une priorité.

Les recherches ont démontré que le langage des signes ne s'arrêtait pas aux seuls gestes des mains, mais également à l'expression du visage ou au contexte de la situation. Certainement qu'en collaborant avec des signeurs, il serait possible d'approfondir la faisabilité d'une telle reconnaissance.

L'analyse d'image ou la reconnaissance vocale dans le domaine de la sécurité doivent être bien étudiés. La redondance des systèmes est la meilleure des sécurités.

Les outils qui ont été développés durant ce travail de diplôme fonctionnent bien. Ils ont leur limite, mais sont fonctionnels. La détection des couleurs est fiable, en partie grâce à la correction de la luminosité. Le repérage du pointeur lumineux est opérationnel. Que ce soit le pointeur laser rouge ou la lampe à DEL, cela fonctionne. La reconnaissance vocale et la synthèse vocale fonctionnent également. Il faut parfois répéter les ordres, mais cela n'est pas trop gênant.

L'interface est simple et compréhensible. L'utilisateur ne se perd pas en ne sachant plus quoi faire. Il a toujours la possibilité de revenir en arrière. L'interface offre également l'accès aux réglages des paramètres des outils.

L'application de démonstration de l'interface est efficace. Elle correspond à une utilisation industrielle réelle. En plus, elle donne envie de s'y intéresser et d'en apprendre plus sur l'interface et les moyens employés.

Je suis très satisfait de terminer ma formation sur un tel travail. J'ai beaucoup appris sur les contraintes liées à la création d'une interface.

CHAPITRE 8

Perspectives

8.1 Améliorations

L'état final de l'interface et du programme au moment de rendre ce travail était satisfaisant, sans pour autant être infaillible. En effet, il comporte certaines erreurs mineures qui pourraient être corrigées avec plus de temps. Ces défauts n'influencent pas directement le résultat du travail de diplôme.

Le programme pourrait également être approfondi afin de le rendre plus souple. Certains points seront d'ailleurs sûrement améliorés d'ici la défense orale de ce travail.

8.2 Utilisations réelles

J'ai essayé de projeter l'interface et ses outils dans des situations réelles. L'exercice n'était pas facile, car en général un produit est développé pour combler un manque ou un besoin. En plus, du style fonction *Pick and Place* du robot Delta, voici quelques idées plausibles.

8.2.1 Gestion de stock

Il faut imaginer un entrepôt de stockage composé d'étagères très hautes. Les caméras de sécurité sont sensées couvrir l'ensemble des étagères. De cette manière, tous les produits stockés sont visuellement accessibles.

Ces produits sont placés dans leur rayon à l'aide d'un robot élévateur.

Afin de sélectionner un produit que nous voudrions descendre, il suffirait de le sélectionner en le pointant à l'aide d'un laser et de donner un ordre vocal adapté.

Ainsi l'utilisateur, n'aura besoin ni de monter, ni d'avoir une télécommande encombrante.

8.2.2 Pilotage de machine de chantier

Un deuxième exemple pratique serait une utilisation des outils développés pour piloter une machine de chantier.

La machine serait équipée de caméras. A l'aide du pointeur laser, l'utilisateur désignerait une zone à creuser, un rocher à déplacer ou encore un endroit où il devrait se déplacer.

Les seuls outils de commande dont devrait s'équiper l'utilisateur seraient un micro, du style oreillette ou intégré au casque de sécurité, et un pointeur laser.

Il serait également possible d'imaginer disposer de deux caméras qui permettraient une estimation de la distance du point à atteindre par trigonométrie.

CHAPITRE 9

Références

- [1] Documentation Demarex S.A. pour Delta95 - V3.3 (2002) par JFS/MHU
- [2] Travail de diplôme 2004 - Application de démonstration pour robot Delta - par M. Christophe Gabbud
- [3] Site web de partage de codes sources en Visual Basic
(<http://www.vbfrance.com>)
- [4] Site web du club des développeurs francophones
(<http://www.developpez.com>)
- [5] Site web de présentation du langage des signes
(<http://www.langage-des-signes.com>)

CHAPITRE 10

Remerciements

Je tiens à remercier les personnes suivantes pour leur disponibilité tout au long du travail de diplôme ainsi que pour les précieux conseils qu'ils m'ont apportés.

- Monsieur Marcuard Jean-Daniel, Professeur à la HES-SO Valais. Pour ses connaissances en robotique ainsi que pour son soutien et l'encadrement durant le projet.
- Monsieur Steiner Amandus, Responsable de l'atelier de productique CR03 à la HES-SO Valais. Pour la mise à disposition d'une place de travail,
- les employés de l'atelier mécanique pour l'exécution rapide des différents éléments constituant la partie mécanique et leurs conseils avisés,
- mon amie Samantha pour son soutien dans toutes les situations,
- mon ami Blaise pour ses conseils en construction mécanique,
- mon ami Fabien pour la relecture et les corrections orthographiques,
- et bien sûr toute ma famille pour leurs encouragements.

Ainsi que toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet.

CHAPITRE 11

Annexes

11.1 Plans du réservoir de chocolats

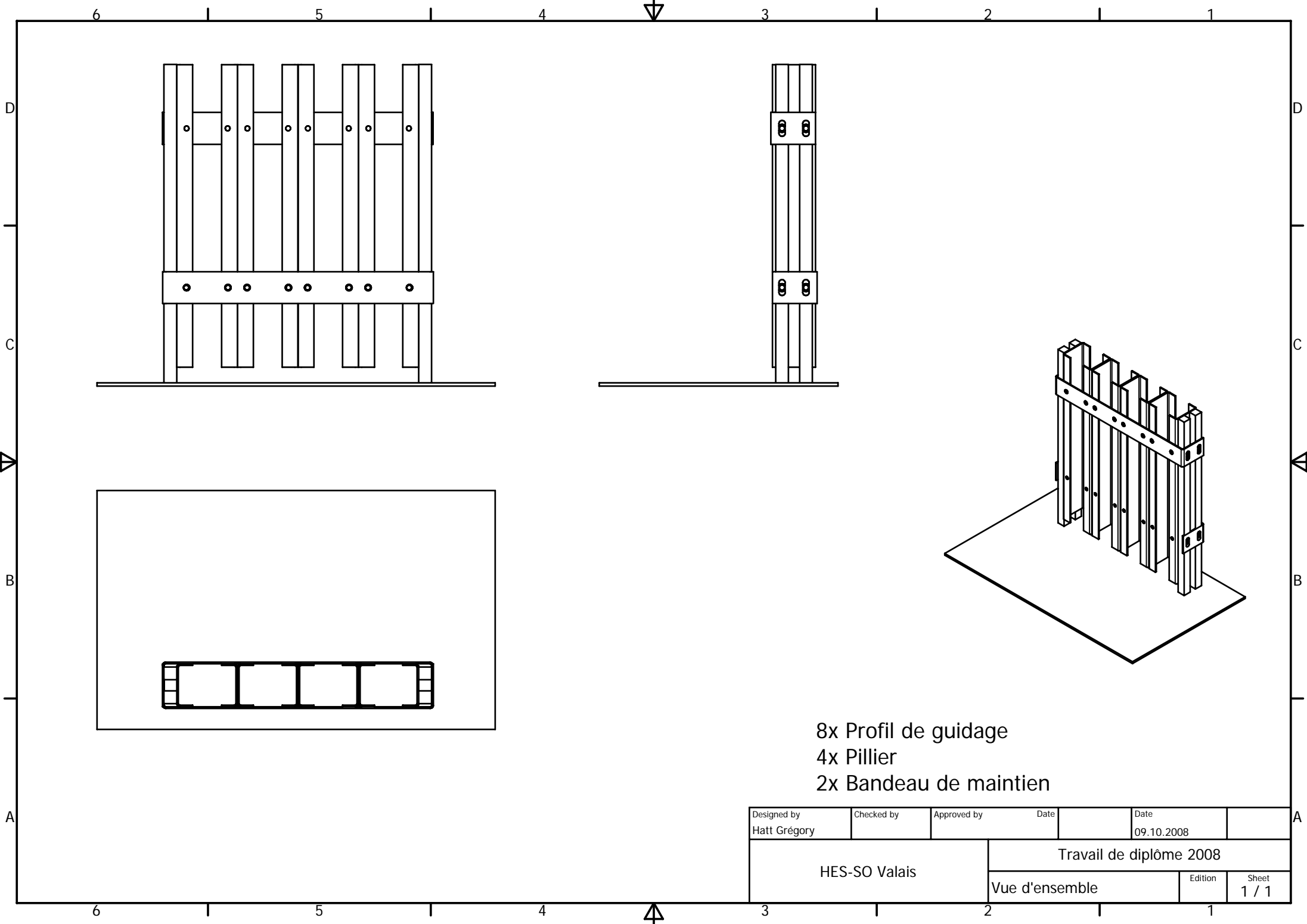
Vue d'ensemble : Représentation générale du réservoir.

Bandeaux de maintien : Bandeaux permettant de maintenir les profils.

Profils de guidage : Profils permettant de guider les chocolats dans le réservoir.

Piliers : Piliers de montage du système.

Râteau d'éjection : Plaque permettant d'éjecter les chocolats du réservoir.

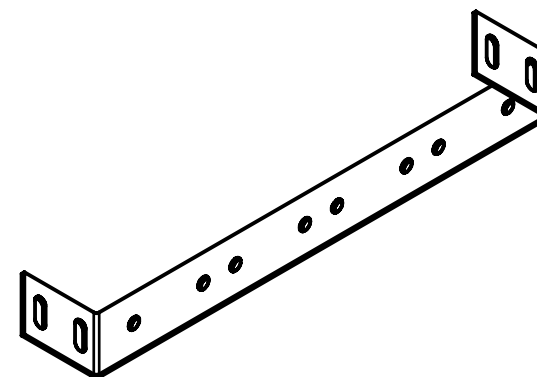
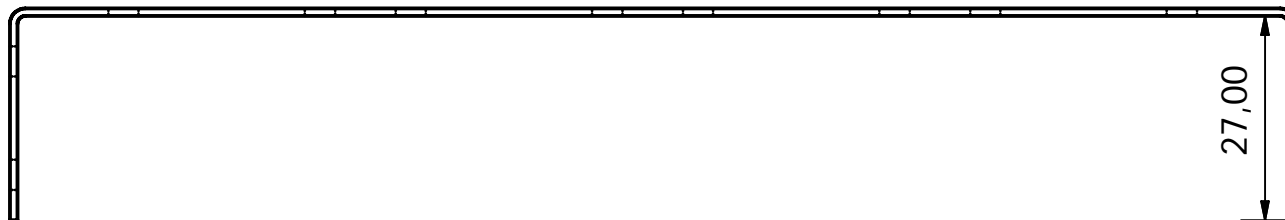
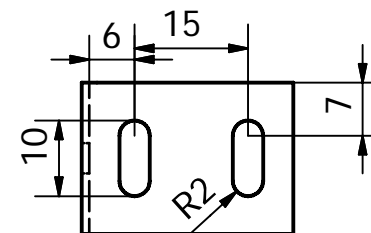
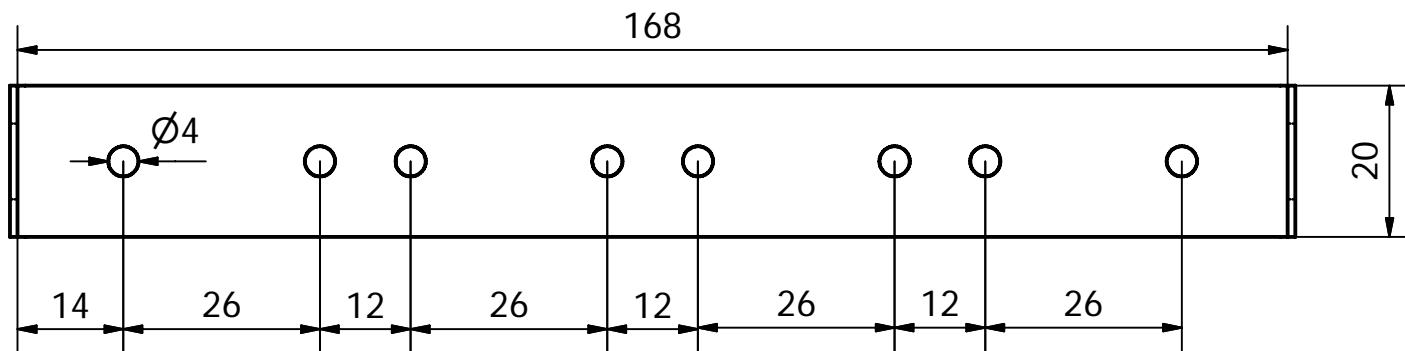


8x Profil de guidage

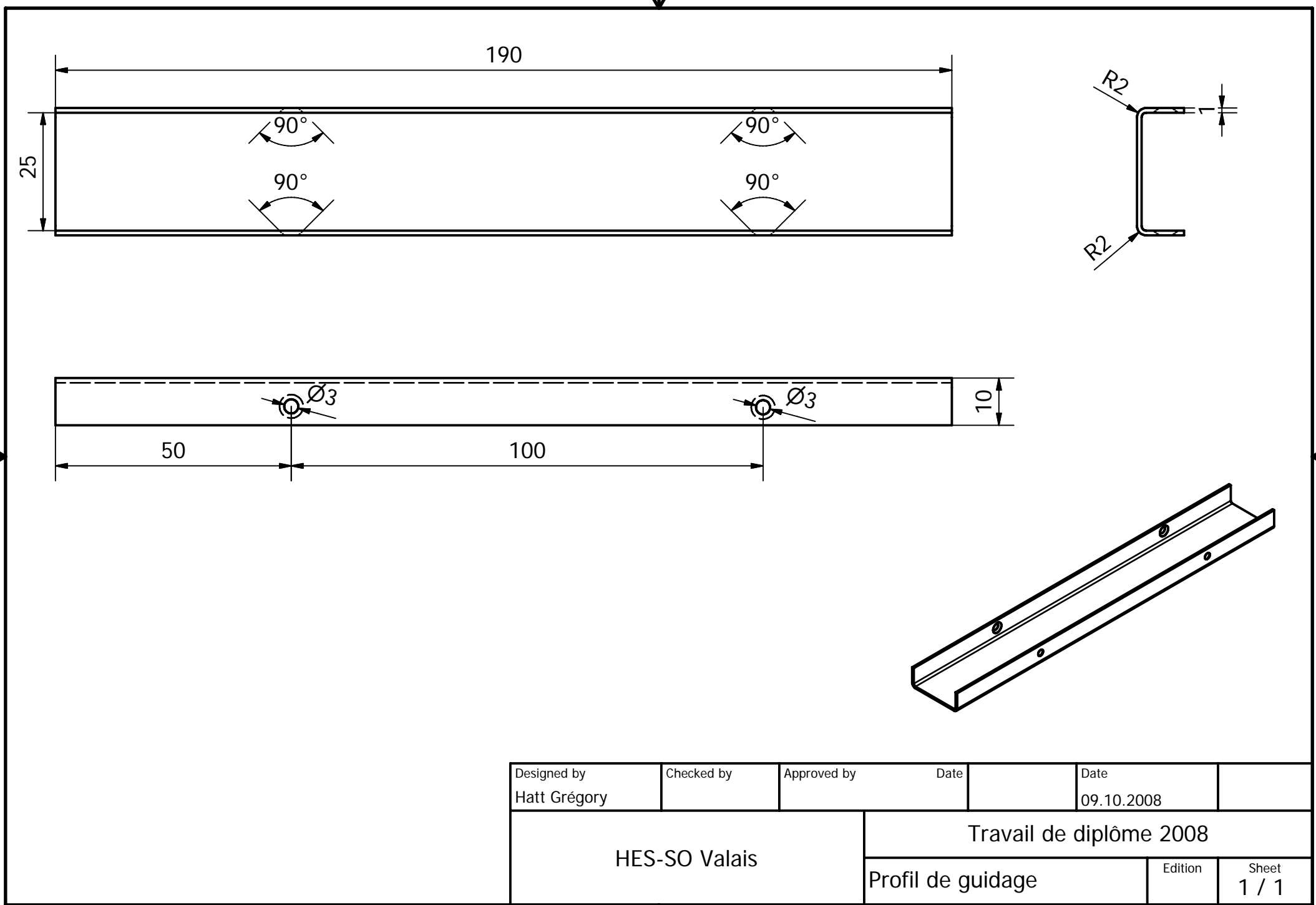
4x Pillier

2x Bandeau de maintien

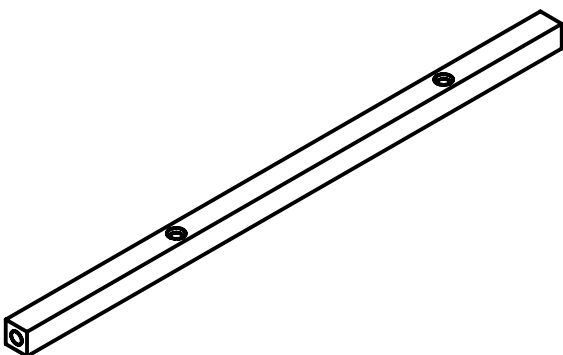
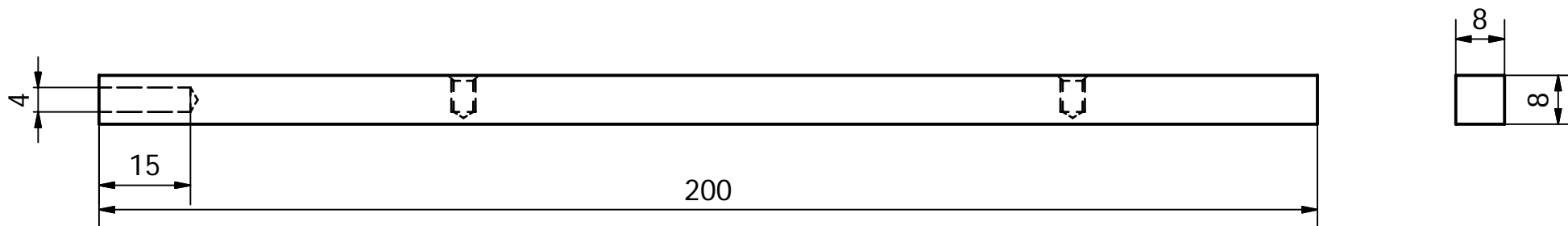
Designed by Hatt Grégory	Checked by	Approved by	Date 09.10.2008	
HES-SO Valais		Travail de diplôme 2008		
		Vue d'ensemble		Edition Sheet 1 / 1



Designed by Hatt Grégory	Checked by	Approved by	Date	Date 09.10.2008	
HES-SO Valais			Travail de diplôme 2008		
			Bandeau de maintien	Edition	Sheet 1 / 1

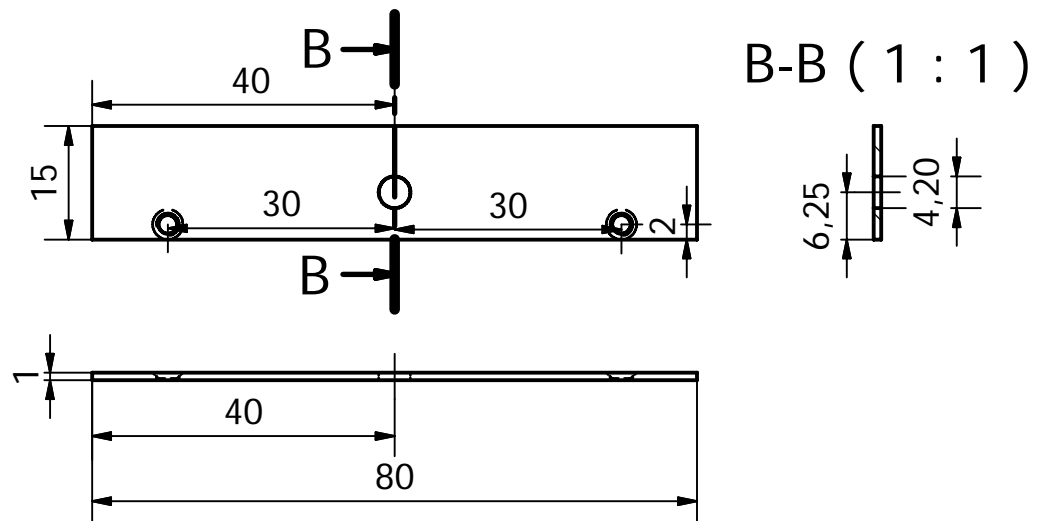


Designed by Hatt Grégory	Checked by	Approved by	Date		Date	09.10.2008	
HES-SO Valais			Travail de diplôme 2008				
			Profil de guidage		Edition	Sheet 1 / 1	



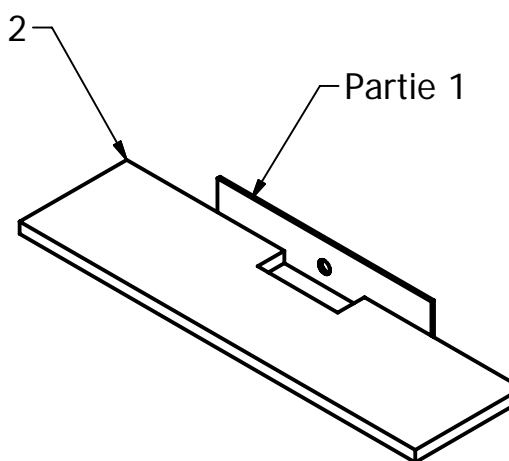
Designed by Hatt Grégory	Checked by	Approved by	Date		Date	15.10.2008	
HES-SO Valais			Travail de diplôme 2008				
			Pilier		Edition	Sheet 1 / 1	

Partie 1

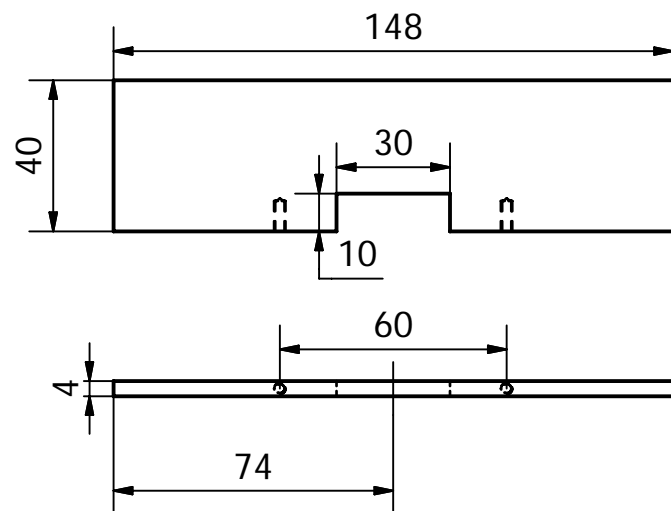


Partie 2

Partie 1



Partie 2



Designed by Hatt Grégory	Checked by	Approved by	Date	Date 15.10.2008	
HES-SO Valais			Travail de diplôme		
Poussoire			Edition	Sheet 1 / 1	

11.2 Contenu du CD

11.2.1 Rapport

11.2.2 Programme de l'interface - Visual Basic

Code source : Ensemble du projet Visual Basic.

Exécutable : Fichier exécutable du programme Visual Basic.

11.2.3 Programme du robot - TurboPascal

Code source : Fichiers sources du programme en TurboPascal.

Fichiers de configuration : Différents fichiers de configuration pour le robot Delta.

11.2.4 Outils Microsoft

actcnc.exe : Moteur de reconnaissance vocale.

speechSDK51.exe : Kit de développement 5.1.

spchcpl.exe : Panneau de configuration du kit de développement 5.1.

AgtX040C_fr.exe : Support de langue française pour la synthèse vocale.

lhttsfrf_fr.exe : Moteur de voix française pour la synthèse vocale.

lhttseng_british.exe : Moteur de voix anglaise pour la synthèse vocale.

11.2.5 Autres

Réservoir : Plans de construction du réservoir.

Médias : Photos et vidéos du travail de diplôme.

Grégory Hatt - 2008

